Efficient Algorithms for Sparse Cyclotomic Integer Zero Testing

Qi Cheng* Sergey P. Tarasov† Mikhail N. Vyalyi‡

October 29, 2008

Abstract

We present two deterministic polynomial time algorithms for the following problem: check whether a sparse polynomial $f(x)$ vanishes at a given primitive nth root of unity ζ_n. A priori $f(\zeta_n)$ may be nonzero and doubly exponentially small in the input size. The existence of a polynomial time procedure in the case of factored n was conjectured by D. Plaisted in 1984, but all previously known algorithms are either randomized, or do not run in polynomial time.

We apply polynomial zero testing algorithms to construct a nondeterministic polynomial time algorithm for the torsion point problem (TP). The problem TP is a particular case of the feasibility problem for a system of polynomial equations in complex numbers (coefficients of polynomials are integers). In the problem TP all coordinates of a solution must be roots of unity.

Key words: algorithm, cyclotomic polynomial, root of unity, sparse representation

1 Introduction

Let $\zeta_n = e^{2\pi i/n}$ be an nth primitive root of unity. *A vanishing sum* of roots of unity has the form

$$\sum_{j=0}^{n-1} a_j \zeta_n^j = 0 \tag{1}$$

where the coefficients a_j are integers.

*School of Computer Science, The University of Oklahoma, Norman, OK 73019, USA. This research is partially supported by NSF Career Award CCR-0237845 of USA and by Project 973 (no: 2007CB807903) of China.

†Dorodnitsyn Computing Center of RAS, Vavilova, 40, Moscow, 119991, Russia. The work is supported by the RFBR grant 08–01–00414.

‡Dorodnitsyn Computing Center of RAS, Vavilova, 40, Moscow, 119991, Russia. The work is supported by the RFBR grants 08–01–00414, 05–01–02803–NTsNIL_a and the grant NS 5294.2008.1.

There are many classification results on vanishing sums of roots of unity. Rédei [17] and Schoenberg [19] described the lattice of coefficients of vanishing sums (see also Rédei [16], de Bruijn [4], Lam and Leung [13]). Conway and Jones [7] gave a lower bound on the size of the support set of a minimal vanishing sum with nonnegative coefficients. The paper by Lam and Leung [13] contains an exact characterization of the set of ℓ^1-norms of vectors of the coefficients of vanishing sums with nonnegative coefficients. Steinberger [21] developed a method for construction of minimal sums with large coefficients.

In this paper we examine the algorithmic aspects of zero testing of sums of roots of unity and consider the following problem. Given an integer n, a finite *support* set J of natural numbers and a set of integer coefficients a_j, $j \in J$ check the equality

$$\sum_{j \in J} a_j \zeta_n^j = 0 \ . \tag{2}$$

Hereafter we call this problem the *cyclotomic test* (CT for brevity), or *sparse cyclotomic integer zero testing*, as $\sum_{j \in J} a_j \zeta_n^j$ is an algebraic integer in the cyclotomic field $\mathbb{Q}(\zeta_n)$.

Due to the irreducibility of cyclotomic polynomials $\Phi_n(x)$, the equality (2) is equivalent to the divisibility of a sparse polynomial

$$f(x) = \sum_{j \in J} a_j x^j \tag{3}$$

by the cyclotomic polynomial $\Phi_n(x)$. Note that $\Phi_n(x) \mid (x^n - 1)$ and it is easy to compute $f(x) \bmod (x^n - 1)$. For this reason hereafter we assume without loss of generality that $\deg f(x) < n$.

A *sparse representation* of a polynomial $f(x) = \sum_{i=0}^{d} a_j x^j$ with integer coefficients is a list of pairs (a_j, j) for $a_j \neq 0$. The length of the list (i.e. the number of nonzero terms in the polynomial) is called *sparseness* and is denoted by $sps(f)$. Integers in a sparse representation are written in binary. The *support* set $\operatorname{supp} a$ of a vector $a = (a_0, \ldots, a_d)^T$ is a set $\{j : a_j \neq 0\}$. The *height* $H(f)$ of f is $\max_{j \in J} |a_j| + 1$. So the size of a sparse representation of a polynomial $f(x)$ is $O(sps(f)(\log H(f) + \log(2 + \deg f)))$.

The decision problem CT is stated formally as follows. The input is a sparse representation of a polynomial $f(x)$ and an integer n written in binary, $\deg f < n$. The output is 'Yes' if $f(\zeta_n) = 0$ and 'No' otherwise.

Our main concern is an efficient (i.e. polynomial time with respect to the input size) algorithm for the problem CT. Note that the sparseness $sps(f)$ and the maximal bit length of integers contained in the input $L = \max(\log H(f), \log(2 + \deg f), \log n)$ do not exceed the input size. We will use the parameters $sps(f)$, L and $\log n$ in bounds of running time below.

1.1 Previous work

The standard algorithm for divisibility of polynomials runs in exponential time w.r.t. the input size of the problem CT. Nonetheless it is shown by Plaisted [15]

that CT is in co-NP (the related problem is called SPARSE-POLY-NONROOT there).[1]

Note that a linear combination of roots of unity with integer coefficients is an algebraic integer. So a straightforward way to check the equality (2) is to compute a rational approximation of its left-hand side and then to compare it with zero. For any $\sigma \in Gal(\mathbb{Q}(\zeta_n)/\mathbb{Q})$, $|\sigma(f(\zeta_n))| \leq sps(f)H(f)$. If $f(\zeta_n)$ is not zero, we have

$$|Norm(f(\zeta_n))| = \prod_{\sigma \in Gal(\mathbb{Q}(\zeta_n)/\mathbb{Q})} |\sigma(f(\zeta_n))| \geq 1. \tag{4}$$

Thus $|f(\zeta_n)| \geq 1/(sps(f)H(f))^{\phi(n)}$, where ϕ is Euler's phi function. This bound is known as the root separation bound. If the bound is close to being tight, it seems that we need exponential precision, i.e. $\phi(n)\log(sps(f)H(f)) = \Omega(n\log(sps(f)H(f))/\log\log n)$, to tell whether a sparse cyclotomic integer is zero or not.

On the other hand, from the inequality (4) one can also conclude that the absolute values of most of the conjugates of a nonzero $f(\zeta_n)$ are not too small. In fact, it can be shown that if we randomly select an element $\sigma \in Gal(\mathbb{Q}(\zeta_n)/\mathbb{Q})$, then with probability at least $1/2$, $|\sigma(f(\zeta_n))| \geq 1/sps(f)H(f)$. So we can perform zero testing of sparse cyclotomic integers in randomized polynomial time. This idea has been used in [5, 2] to design randomized algorithm for polynomial identity testing and zero testing of expressions involving roots of rationals.

Note that in some cases a large conjugate can be found deterministically. (See Theorem 2 in [6].)

To our knowledge the best deterministic zero testing algorithm prior to our extended abstracts [6, 23] was developed by Filaseta and Schinzel (see [10, Theorem 3]). It runs in subexponential time, provided the prime power decomposition of n is given. More precisely, the estimate of the running time in [10] contains a factor 2^s where s is the number of prime divisors of n. The algorithm is based on the observation that $f(\zeta_n) = 0$ iff

$$x^n - 1 \ \text{ divides } \ f(x)\prod_{p|n}(x^{n/p} - 1),$$

also observed by Plaisted in [15].

In fact, Filaseta and Schinzel proposed in [10] an algorithm for a related problem: to check whether a sparse polynomial $f(x)$ is divisible by some cyclotomic polynomial. In other words, whether there exists n such that $(n, f(x))$ is a positive instance of the CT problem. We call this problem the general cyclotomic test (GCT for brevity). The running time of the GCT-algorithm in [10] is subexponential, it uses as a subroutine the aforementioned subexponential

[1] There is even a more resolute statement in [15, p. 132]: "The author believes he has a method for solving SPARSE-POLY-NONROOT in polynomial time if the prime factorization of M is given." To our knowledge this result is unpublished.

cyclotomic test. It's worth noting that for a fixed sparseness this algorithm runs in polynomial time.

Adding an existential quantifier usually makes a problem harder. Indeed, the GCT is NP-hard. This result is implicitly contained in Plaisted's theorem [15, Theorem 5.1][2].

The result of Plaisted that CT $\in$ co-NP mentioned above implies GCT $\in \Sigma_2$.

A more sophisticated algorithm for GCT is described in a recent paper by Filaseta, Granville, and Schinzel [9]. However, it uses the same subexponential cyclotomic test. So, this algorithm cannot be applied to prove that GCT $\in$ NP.

Another type of problem related to the cyclotomic tests are specific cases of the *complex feasibility problem* $\mathrm{FEAS}_{\mathbb{C}}$. The problem is to verify the satisfiability of a system of polynomial equations in complex numbers (coefficients of polynomials are integers). If the system includes equations $x_i^{d_i} - 1 = 0$ for each variable x_i then the coordinates of all solutions are roots of unity. This specific case of the problem $\mathrm{FEAS}_{\mathbb{C}}$ is called the torsion point problem (TP for brevity)[3]. It was studied by Plaisted [15] for the univariate case (the TP_1 problem for brevity[4]) and by Rojas [18] for the multivariate case. Plaisted thus proved implicitly that TP_1 is NP-hard.

Koiran proved in [12] that $\mathrm{FEAS}_{\mathbb{C}} \in \mathcal{AM}$ under the Generalized Riemann Hypothesis. Of course, the same inclusion holds for the TP problem. Rojas [18] improved this result for the TP problem in various ways: TP $\in \mathcal{AM}$ under a weaker number-theoretic hypothesis and $\mathrm{TP}_1 \in \mathrm{NP}^{\mathrm{NP}}$ unconditionally. Also, he proved that for a fixed number of variables and fixed degrees of roots of unity the TP problem is in P.

Rojas indicated that the TP problem looks more tractable than the general $\mathrm{FEAS}_{\mathbb{C}}$ problem and conjectured that TP $\in$ NP, which is unlikely for the $\mathrm{FEAS}_{\mathbb{C}}$ problem.

1.2 Our results

Our contribution is polynomial time deterministic algorithms for the cyclotomic test in the case of a general (not factored) n. As a direct consequence we show that the GCT and TP problems are in NP.

Theorem 1. CT $\in$ P.

Theorem 2. GCT $\in$ NP.

Theorem 3. TP $\in$ NP.

It follows also from previous results (Plaisted [15], Theorem 5.1 for GCT and Theorem 3.3 for TP_1) and Theorems 2 and 3 that the problems GCT and TP (and even TP_1) are NP-complete.

[2] Theorem 5.1 in [15] speaks about complex numbers of modulus 1 but its proof is also valid for the roots of unity. We are grateful to an unknown referee who explained this fact to us.

[3] This potential application for our methods was indicated by one of the referees.

[4] Note a difference between the problems GCT and TP_1: the former does not specify a root of unity at all and the latter indicates the degree of a root.

Two efficient cyclotomic tests were proposed in conference papers [6, 23]. They use different techniques and their algorithmic behavior is also different. The matrix multiplication algorithm from [23] computes a matrix M of size $sps(f) \times sps(f)$ and a vector $\tilde{f}$ of dimension $sps(f)$. Then it checks the equality $M\tilde{f} = 0$. The recursive algorithm from [6] applies the divide-and-conquer approach and reduces an instance of the CT problem to a number of smaller instances.

It is worth noting that the algorithms share some common features. They use a *partial prime decomposition* to avoid factorization of n. Also they can be expressed in terms of operations with sparse vectors of exponentially large dimension. More exactly, the algorithms can be applied to a more general problem, which we call *cyclotomic array testing* (CAT) see Section 4.

The paper is organized as follows. Section 2 contains an exposition of the matrix multiplication algorithm. Section 3 presents the recursive algorithm. In Section 4 we introduce the cyclotomic array testing problem and modify our algorithms to solve it. Section 5 contains the proof of Theorem 3. In the final Section 6 we discuss open questions related to zero testing.

2 Matrix multiplication algorithm

We start from an informal outline of the algorithm.

To solve an instance $(n, f(x))$ of the CT problem one checks the equality $f(\zeta_n) = 0$. Recall that we restrict the problem to the case $\deg f < n$. The polynomials of degree $< n$ form a linear space. Polynomials that vanish at ζ_n form a subspace of this space (*the space of vanishing sums*). So one can regard the CT problem as a specific case of typical computational linear algebra problem: does a vector belong to a subspace? Standard linear algebra techniques are too expensive to work here as the vector and the space involved have exponential dimension w.r.t. the input size.

Nevertheless, it is possible to reduce dimensions. This reduction is based on a specific structure of the space of vanishing sums. Namely, the space of polynomials of degree $< n$ admits a structure of a tensor product of polynomially many spaces of polynomial dimension and the space of vanishing sums has a convenient description in terms of this tensor product (see the exact statements, especially Theorem 4, in Subsection 2.1). This description is well-known (it appears in different forms in [13, 19, 21]). We need to reformulate this description in order to characterize the space of vanishing sums as the kernel of a tensor factored operator (see Subsection 2.2).

To avoid factorization of n we need to modify the operator taking into account the sparseness of the polynomial f. This argument is explained in Subsection 2.3 (see Lemma 6). Yet all these steps do not change the dimensions. A way to truncate dimensions is based on a simple technical trick which is also explained in Subsection 2.3.

Finally, in Subsection 2.4 we give a description of the matrix multiplication algorithm.

The algorithm admits variations by changing operators in the tensor product. We adopt a choice corresponding to an earlier version of the algorithm (see [23]). Thus the algorithm description in [23] coincides with the description in Subsection 2.4.

2.1 Space of vanishing sums

Let

$$n = p_1^{t_1} \cdot p_2^{t_2} \cdot \ldots \cdot p_r^{t_r} \tag{5}$$

be the prime power decomposition of n. For a positive integer n let V_n be the group $\mathbb{Q}$-algebra for the cyclic group of order n. Basically, we will consider V_n as an n-dimensional vector space over the field $\mathbb{Q}$ of rationals equipped with the canonical basis $\{e_0, e_1, \ldots, e_{n-1}\}$. Sometimes we will identify V_n with the quotient ring $\mathbb{Q}[t]/(t^n - 1)$ (see Lemma 1 below). In these cases we assume an isomorphism between V_n and $\mathbb{Q}[t]/(t^n - 1)$ that maps e_j to t^j.

Let $\varphi\colon V_n \to \mathbb{Q}[\zeta_n]$ be an evaluation map — a $\mathbb{Q}$-linear map acting on the basis vectors by the rule

$$\varphi(e_k) = \zeta_n^k \ . \tag{6}$$

We use notation X_n for the kernel of the evaluation map φ and call X_n *the space of vanishing sums*. This space is directly related to the CT problem: by definition, $f(\zeta_n) = 0$ iff $f(x) \in X_n$ provided $\deg f < n$.

Thus, to solve the problem CT it is sufficient to check that the polynomial $f(x)$ belongs to the space of vanishing sums. This approach does not require any approximation of the value $f(\zeta_n)$. Instead, we need a technique to operate with vectors of polynomially bounded sparseness in a space of exponentially large dimension. For this purpose we use a representation of V_n as a tensor product of polynomially many factors such that the space of vanishing sums has a nice description in terms of this tensor decomposition (see Theorem 4).

At first, we recall the well-known characterization of the space of vanishing sums.

Lemma 1 ([17]). *X_n is spanned by polynomials*

$$x^j \frac{x^n - 1}{x^{n/p} - 1}, \tag{7}$$

where p runs over all prime divisors of n and $0 \le j < n/p$.

Proof. The cyclotomic polynomial $\Phi_n(x)$ is the greatest common divisor of polynomials

$$\frac{x^n - 1}{x^{n/p} - 1}$$

(any non-primitive root ζ_n^j is a root of some polynomial $x^{n/p} - 1$ where p is a common prime divisor of n and j). □

To obtain a compact form for the generators (7) we use a tensor decomposition of V_n in the form

$$V_n \cong \bigotimes_{k=1}^{r} V_{p_k} \otimes V_{n/P} \tag{8}$$

where $P = p_1 p_2 \dots p_r$. This decomposition was introduced by Lam and Leung [13] (see also [21]). It can be defined as follows. The isomorphism maps a vector e_j from the canonical basis to the tensor product of basis vectors:

$$e_j \mapsto e_{j_1} \otimes e_{j_2} \otimes \dots \otimes e_{j_r} \otimes e_{j'}. \tag{9}$$

In (9) $j' = j \bmod n/P$ and j_k is the t_k-th digit in p_k-ary representation of j:

$$j = j_0^{(k)} + j_1^{(k)} p_k^1 + \dots + j_{t-1}^{(k)} p_k^{t-1} + \dots, \quad j_k = j_{t-1}^{(k)}. \tag{10}$$

Example 1. Let $n = 30 = 2 \cdot 3 \cdot 5$ and $j = 9$. Then $j_1 = 1$, $j_2 = 0$, $j_3 = 4$, $j' = 0$. Note that for a square-free n we have $j_k = j \bmod p_k$. In this case the last factor in the decomposition (8) is 1-dimensional so it can be omitted.

Example 2. Let $n = 144 = 2^4 3^2$ and $j = 15$. Then $j_1 = 1$, $j_2 = 2$ ($15 = 0 + 2 \cdot 3 + 1 \cdot 3^2$), $j' = 15 \bmod 24 = 15$.

To check that (9) defines an isomorphism we apply the following lemma.

Lemma 2. *The mapping $\iota\colon j \mapsto (j_1, \dots, j_k, j')$ is a bijection of the set $\{0, \dots, n-1\}$ onto*

$$\{0, \dots, p_1 - 1\} \times \dots \times \{0, \dots, p_r - 1\} \times \{0, \dots, n/P - 1\}.$$

Moreover, both maps ι and ι^{-1} can be computed in polynomial time provided the factorization of n is known. The map ι can be computed in time $O(\log^3 n)$ and the inverse map ι^{-1} can be computed in time $O(\log^4 n)$.

Proof. Let us describe the inverse map ι^{-1}. The numbers j_k and $j' \bmod p_k^{t_k - 1}$ determine the residue of j modulo $p_k^{t_k}$:

$$j \bmod p_k^{t_k} = j_k p_k^{t_k - 1} + j' \bmod p_k^{t_k - 1}. \tag{11}$$

By the Chinese remainder theorem $j \bmod n$ is determined by residues modulo $p_k^{t_k}$.

Applying efficient algorithms for modular arithmetic (see, e.g., [3]) we get the second statement of the lemma.

Note that the t_k-th digit in p_k-ary representation of n can be computed by the Horner scheme using $O(t_k)$ arithmetic operations. Since

$$2^{\sum_k t_k} \le \prod_k p_k^{t_k} = n,$$

the overall number of arithmetic operations is $O(\log n)$. All operations are applied to $(\log n)$-bit integers. So, a division takes $O(\log^2 n)$ time. Thus, computation of the map ι takes $O(\log^3 n)$ time.

To compute the inverse map ι^{-1} one should compute the residues $j \bmod p_k^{t_k}$ using (11) and apply the algorithm reconstructing j by these residues. The first step takes $O(r(\log\log n)^2)$ arithmetic operations with $O(\log n)$-bit integers. The second can be done by r applications of the extended Euclid algorithm. Each application takes a time $O(\log^3 n)$. Since $r \le \log n$, we get the time bound $O(\log^4 n)$ for the computation of the inverse map ι^{-1}. □

Note that the isomorphism (9) maps vectors of the canonical basis $\{e_j\}$ of V_n to the vectors of the canonical basis of the tensor product in the right-hand side of (8). Applying it to a vector of sparseness m we obtain a vector of the same sparseness in the tensor product. Lemma 2 implies that this transformation can be done efficiently.

To describe X_n in terms of tensor decomposition (8) we define the vectors $\hat{1}_p \in V_p$ by

$$\hat{1} = \sum_{j=0}^{p-1} e_j. \tag{12}$$

The next theorem is a reformulation of Lemma 1.

Theorem 4. *$X_n = \operatorname{Ker}\varphi$ is a sum of subspaces X_n^k, where*

$$\begin{aligned} &X_n = X_n^1 + X_n^2 + \cdots + X_n^r, \\ &X_n^k = V_{p_1} \otimes \cdots \otimes V_{p_{k-1}} \otimes \mathbb{Q}\hat{1} \otimes V_{p_{k+1}} \otimes \cdots \otimes V_{p_r} \otimes V_{n/P}, \quad 1 \le k \le r \ . \end{aligned} \tag{13}$$

(Hereafter we identify X_n and its image by isomorphism (8).)

A stronger form of Theorem 4 is contained in the paper by Lam and Leung [13, Theorem 2.2]. (They attribute the theorem to Rédei, de Bruijn and Schoenberg.)

Proof. The exponents of non-zero terms in a generator

$$f_{j,k} = x^j \frac{x^n - 1}{x^{n/p_k} - 1}$$

form an arithmetic progression modulo n:

$$j,\ j + \frac{n}{p_k},\ \ldots, j + a\frac{n}{p_k}, \ldots \quad (a = 0, \ldots, p_k - 1).$$

Note that an addition of n/p_k does not change j' and j_s for $s \ne k$ since it does not change the residue modulo $p_s^{t_s}$. So, the k-th components of $\iota(j)$ also form an arithmetic progression

$$j_k, j_k + b,\ \ldots, j_k + ab, \ldots \quad (a = 0, \ldots, p_k - 1),$$

where b is a residue of $n/p_k^{t_k}$ modulo p_k. This residue is non-zero. Thus the k-th component takes all possible values. In terms of tensor decomposition this means that $f_{j,k}$ can be written as

$$e_{j_1} \otimes \cdots \otimes e_{j_{k-1}} \otimes \hat{1} \otimes e_{j_{k+1}} \otimes \cdots \otimes e_{j_r} \otimes e_{j'},$$

thus immediately implying (13). □

2.2 Kernel representation of the space of vanishing sums

To use Theorem 4 in the algorithm we rewrite (13) representing the space of vanishing sums as the kernel of a suitable operator. The decomposition (13) suggests the form of the operator as a tensor product of operators acting on tensor factors of the decomposition (8).

For exact statements we need a bit of tensor linear algebra.

Lemma 3. *Let $\sum_k u_k \otimes v_k = 0$ and vectors u_k are linearly independent. Then $v_k = 0$ for any k.*

This simple and useful fact implies the following lemma.

Lemma 4. *Let $A\colon U \to U'$, $B\colon V \to V'$ be linear operators such that $\operatorname{Ker} A = 0$, $\operatorname{Ker} B = 0$. Then $\operatorname{Ker}(A \otimes B) = 0$. For arbitrary operators A, B we have $\operatorname{Ker}(A \otimes B) = \operatorname{Ker} A \otimes V + U \otimes \operatorname{Ker} B$.*

Applying Lemma 4 inductively we obtain the expected form of the kernel of a tensor product of operators. Let I_n be the indentity operator on the space V_n. (Recall that $\dim V_n = n$.)

Lemma 5. *Let $A = \otimes_{k=1}^{r} A_k \otimes I_{n_{r+1}}$ be a tensor product of operators A_k acting on a space $\otimes_{k=1}^{r+1} V_{n_k}$. Then*

$$\operatorname{Ker} A = \sum_k V_{n_1} \otimes \cdots \otimes \operatorname{Ker} A_k \otimes \cdots \otimes V_{n_r} \otimes V_{n_{r+1}}. \tag{14}$$

Comparing (14) to (13) we conclude that the space of vanishing sums is the kernel of a tensor product

$$A = \bigotimes_{k=1}^{r} A_k \otimes I \tag{15}$$

for any set of operators A_k such that $\operatorname{Ker} A_k = \mathbb{Q}\hat{1}_{n_k}$. Note that in this case $n_k = p_k$ for $k \le r$ and $n_{r+1} = n/P$.

2.3 Using sparseness

Zero testing is thus reduced to checking that $Af = 0$, where A is defined by (15). Until this point, the check may seem quite inefficient. To perform the check it appears that one should operate in a space of exponentially large dimension and use the prime power decomposition of n. To overcome both difficulties we take into account the sparseness of f.

First, we will get rid of the complete factorization. We will instead use a partial prime decomposition

$$n = p_1^{t_1} \dots p_\ell^{t_\ell} q \ , \tag{16}$$

where p_k are the prime divisors of n less than $sps(f) + 1$. Note that for an instance of the problem CT the decomposition (16) can be computed efficiently as p_k are upperbounded by the input size.

Lemma 6. *In the notation above $Af = 0$ iff $A'_\ell f = 0$ where*

$$A'_\ell = \bigotimes_{k=1}^{\ell} A_k \otimes I_N, \quad \text{where } N = \frac{n}{\prod_{i=1}^{\ell} p_i} \ .$$

The proof of Lemma 6 is by induction using the following observation.

Lemma 7. *Let $f \in \operatorname{Ker} A'_\ell$. Expand f as a combination of the canonical basis vectors*

$$f = \sum_{J=(j_1,\ldots,j_r;j_{r+1})} a_J e_J, \quad \text{where } e_J = e_{j_1} \otimes \cdots \otimes e_{j_\ell} \otimes \cdots \otimes e_{j_{r+1}}. \tag{17}$$

Let S be the set of all possible values of j_ℓ in the expansion (17)*. If $|S| < n_\ell$ then $f \in \operatorname{Ker} A'_{\ell-1}$.*

Proof. Group terms of the expansion (17) with respect to the value of ℓ-th factor:

$$f = \sum_{s \in S} f_s \otimes e_{i_s} \otimes e_{\bar{j_s}}.$$

Here $f_s \in \otimes_{k=1}^{\ell-1} V_{n_k}$.

From $\operatorname{Ker} A_\ell = \mathbb{Q}\hat{1}_{n_\ell}$ we conclude that the set $\{A_\ell e_{i_s}\}_{s \in S}$ consists of linearly independent vectors as well as the set $\{A_\ell e_{i_s} \otimes e_{\bar{j_s}}\}_{s \in S}$. Lemma 3 shows that $A' f_s = 0$ for any s where $A' = \bigotimes_{k=1}^{\ell-1} A_k$. Thus $A'_{\ell-1} f = (A' \otimes I_{n_\ell}) f = 0$. □

In order to avoid operations with exponentially long vectors in testing the equality $Af = 0$ we restrict the operator A on the coordinate subspace spanned by the canonical basis vectors from the support of the vector f.

By $\tilde{f}$ we denote a vector produced from f by removing zero components and by $\tilde{A}$ we denote a matrix formed by columns of matrix A indexed by elements from the support of f (see Fig. 1(a)). Conditions $Af = 0$ and $\tilde{A}\tilde{f} = 0$ are equivalent by construction. It is a well-known fact from linear algebra that the latter condition is equivalent to $\tilde{A}^T \tilde{A} \tilde{f} = 0$.

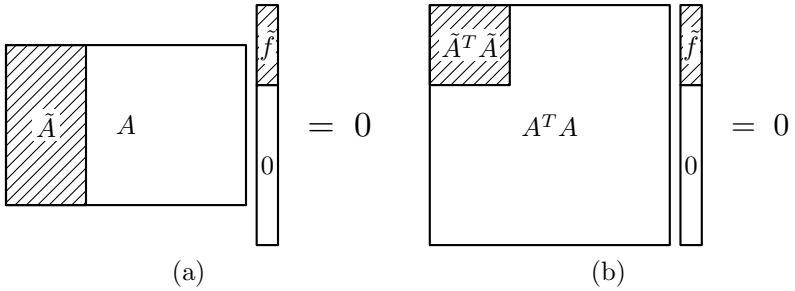

Figure 1: Truncating dimension

Note that $\tilde{A}^T\tilde{A}$ is a $(sps(f)\times sps(f))$ minor of the matrix A^TA (see Fig. 1(b)). For A'_ℓ introduced in Lemma 6 it is easy to express a matrix element of ${A'}_\ell^T A'_\ell$ in terms of matrix elements of A_k. Indeed,

$$ {A'}_\ell^T A'_\ell = \bigotimes_{k=1}^{\ell} A_k^T A_k \otimes I_N, \quad \text{where } N = \frac{n}{\prod_{i=1}^{\ell} p_i} \ . $$

To compute a matrix element with indices $(j'_1,\ldots,j'_\ell,j'_{\ell+1})$ and $(j''_1,\ldots,j''_\ell,j''_{\ell+1})$ one can compute the product of matrix elements

$$ \prod_{k=1}^{\ell} \left(A_k^T A_k\right)_{j'_k,j''_k} \delta(j'_{\ell+1},j''_{\ell+1}). $$

2.4 Description of the matrix multiplication algorithm

The algorithm computes the vector ${\tilde{A}'}_\ell^T \tilde{A}'_\ell \tilde{f}$ of dimension $sps(f)$ and compares it with zero vector.

There is a freedom in choice of operators A_k. To be in accordance with the earlier version of the algorithm (see [23]) we choose operators A_k defined by the following $(n_k-1)\times n_k$-matrices

$$ \begin{pmatrix} 1 & -1 & 0 & \ldots & 0 \\ 1 & 0 & -1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & 0 & 0 & \ldots & -1 \end{pmatrix} \tag{18} $$

It is clear that $\operatorname{Ker} A_k = \mathbb{Q}\hat{1}_{n_k}$ and

$$ A_k^T A_k = \begin{pmatrix} n_k-1 & -1 & -1 & \ldots & -1 \\ -1 & 1 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ -1 & 0 & 0 & \ldots & 1 \end{pmatrix} \tag{19} $$

A detailed description of the algorithm is presented in Fig. 2. Correctness of the algorithm follows from the above consideration. Let's estimate the running time of the algorithm. The preprocessing steps 1–3 can be done by $O(k\log n)$ arithmetic operations with $O(\log n)$ integers. By Lemma 2 the step 4 takes a time $O(k\log^3 n)$. On the step 5 $O(k^2\log n)$ arithmetic operations are performed. So, this step takes a time $O(k^2\log^3 n)$. On the final steps 6–7 $O(k^2)$ arithmetic operations with $O(\max(\log n,\log H(f))$-bit integers are performed. So, these steps take a time $O(k^2\max^2(\log n,\log H(f)))$.

Thus, the overall time bound for the matrix multiplication algorithm is $O(sps^2(f)L^3)$, where $L=\max(\log n,\log H(f))$.

Input: an integral polynomial $f(x)$ given in sparse form and an integer n. The degree of f is less than n.
Output: "Yes" if $f(\zeta_n) = 0$, "No" otherwise.

1. If $f(x)$ is a zero polynomial, then return "Yes".

2. Let k be the sparseness of $f(x)$. Write $n = p_1^{t_1} p_2^{t_2} \cdots p_l^{t_l} q$, where $p_1, p_2, \cdots, p_l$ are primes less than $k+1$ and q does not have prime factors less than $k+1$.

3. Compute $P = p_1 \cdot \ldots \cdot p_l$.

4. For each j from the support set of f compute the index list $(j_1, j_2, \ldots, j_l, j_{l+1})$ by the rules

$$j_m \leftarrow t_m\text{-th digit in } p_m\text{-ary representation of } j \text{ for } m \le l,$$
$$j_{l+1} \leftarrow j \bmod n/P.$$

5. Compute matrix elements of a $k \times k$ matrix M using the index lists by the rule

$$M(j', j'') \leftarrow \delta(j'_{\ell+1}, j''_{\ell+1}) \prod_{s=1}^{l} m_s,$$
$$m_s \leftarrow \begin{cases} p_s, \text{ if } j'_s = j''_s = 0, \\ 1, \text{ if } j'_s = j''_s \neq 0, \\ -1, \text{ if } j'_s = 0 \text{ and } j''_s \neq 0 \text{ or } j'_s \neq 0 \text{ and } j''_s = 0, \\ 0 \text{ otherwise.} \end{cases}$$

6. Compute vector b of dimension k by the rule

$$b(j) \leftarrow \sum_{m \in \operatorname{supp} f} M(j, m) f_m.$$

Here j, m run over the support set of f and f_m is the m-th coefficient of f.

7. If all components of b are zero then return "Yes". Otherwise return "No".

Figure 2: Matrix multiplication algorithm for zero testing

3 Recursive algorithm

In this section we present a recursive algorithm for the cyclotomic test. First observe that if n is a prime and $f(x)$ is a nonzero integral polynomial of sparseness less than n, then $f(\zeta_n)$ cannot be zero. This fact can be derived from the following theorem originally due to Chebotarev.

Proposition 1. If n is a prime, then any minor of the matrix $(\zeta_n^{ij})_{1\leq i,j\leq n}$ is not zero.

There are many proofs of the Chebotarev theorem. For an elementary one, see [22]. By studying selected minors of the matrix $(\zeta_n^{ij})_{1\leq i,j\leq n}$ when n is not a prime, we show that if f is a nonzero integral polynomial and all the prime factors of n are greater than $sps(f)$, then the cyclotomic integer $f(\zeta_n)$ can not be zero. If n has small prime factors, then from a sparse cyclotomic integer $f(\zeta_n)$, our algorithm produces a list of sparse cyclotomic integers in *smaller* field, such that $f(\zeta_n)$ is zero iff all the elements in the list are zero. The algorithm applies the procedure recursively on each cyclotomic integer in the list until we reach a field where the zero testing problem can be easily solved. The recursion can have many levels. As the recursion goes deeper, the number of cyclotomic integers increases, and in some cases, the sum of their sparseness also increases, nonetheless we are able to show that the algorithm runs in polynomial time.

3.1 Key lemmas for derandomization

It is well known that the ring of integers in cyclotomic field $\mathbb{Q}(\zeta_n)$ consists of all the elements in $\mathbf{Z}[\zeta_n]$. The field automorphism of $\mathbb{Q}(\zeta_n)$ is isomorphic to $(\mathbf{Z}/n\mathbf{Z})^*$. For an integer $i \in (\mathbf{Z}/n\mathbf{Z})^*$, let $\sigma^{(i)}$ denote the field automorphism which sends ζ_n to ζ_n^i. Then for any integral polynomial f, we have

$$\sigma^{(i)}(f(\zeta_n)) = f(\zeta_n^i).$$

First we prove a general lemma

Lemma 8. *Let E be a subfield of F. Let $\alpha_1, \alpha_2, \cdots, \alpha_k$ be elements in F. If there exist k field automorphisms $\sigma_1, \sigma_2, \cdots, \sigma_k \in Gal(F/E)$ such that the matrix*

$$V = \begin{pmatrix} \sigma_1(\alpha_1) & \sigma_1(\alpha_2) & \cdots & \sigma_1(\alpha_k) \\ \sigma_2(\alpha_1) & \sigma_2(\alpha_2) & \cdots & \sigma_2(\alpha_k) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_k(\alpha_1) & \sigma_k(\alpha_2) & \cdots & \sigma_k(\alpha_k) \end{pmatrix}$$

is nonsingular, then $\alpha_1, \alpha_2, \cdots, \alpha_k$ are linearly independent over E .

Proof. Suppose that $\alpha_1, \alpha_2, \cdots, \alpha_k$ are linearly dependent over E. Then there exist $a_1, a_2, \cdots, a_k \in E$ such that $\sum_{i=1}^k a_i\alpha_i = 0$ and $a_i \neq 0$ for at least one i. Hence $\sigma_j(\sum_{i=1}^k a_i\alpha_i) = \sum_{i=1}^k a_i\sigma_j(\alpha_i) = 0$ for all $1 \leq j \leq k$. This means that

the vectors

$$\begin{pmatrix} \sigma_1(\alpha_1) \\ \sigma_2(\alpha_1) \\ \vdots \\ \sigma_k(\alpha_1) \end{pmatrix}, \begin{pmatrix} \sigma_1(\alpha_2) \\ \sigma_2(\alpha_2) \\ \vdots \\ \sigma_k(\alpha_2) \end{pmatrix}, \cdots, \begin{pmatrix} \sigma_1(\alpha_k) \\ \sigma_2(\alpha_k) \\ \vdots \\ \sigma_k(\alpha_k) \end{pmatrix}$$

are linearly dependent over $E \subseteq F$. Thus the matrix V is singular, which leads to a contradiction. □

Let k be positive integers and f be an integral polynomial given in sparse form with $sps(f) = k$. Write $n = p_1^{\beta_1} p_2^{\beta_2} \cdots p_l^{\beta_l} r$, where $p_1, p_2, \cdots, p_l$ are distinct primes less than $k+1$ and r is free of prime factors less than $k+1$. Note that it may be hard to factor r. As observed in [15, 10], $f(\zeta_n) = 0$ iff

$$x^n - 1 \ \text{ divides } \ f(x) \prod_{p|n} (x^{n/p} - 1).$$

If the expansion of the latter polynomial has a short sparse representation, then we can check quickly whether $x^n - 1$ divides it or not by replacing x^e in the expansion with $x^{e \mod n}$ and testing whether we have a zero polynomial or not. Thus if $r = 1$ and $l \leq 2$, then we can solve the zero testing problem of cyclotomic integers efficiently.

For $q \in \{p_1, p_2, \cdots, p_l, r\}$, since $\zeta_n^e = \zeta_n^{aq+b} = (\zeta_n^q)^a \zeta_n^b$ where a and b are quotient and remainder respectively of division of e by q, we can write $f(\zeta_n)$ in the following form

$$g_t(\zeta_n^q)\zeta_n^{e_t} + g_{t-1}(\zeta_n^q)\zeta_n^{e_{t-1}} + \cdots + g_1(\zeta_n^q)\zeta_n^{e_1} \tag{20}$$

such that exponents $e_t, e_{t-1}, \cdots, e_1$ fall in t different classes modulo q, and $g_i(x)$'s are sparse polynomials. We divide the zero testing problem of (20) into three cases:

1. $gcd(q, n/q) = 1$ and $t < q$, which includes the case that $q = r$; or
2. $gcd(q, n/q) = 1$ and $t = q$, which implies that q is a prime; or
3. $gcd(q, n/q) > 1$, which implies that $q^2 | n$.

Each case will be handled by one of the following lemmas.

Lemma 9. *If $t < q$ and $gcd(q, n/q) = 1$, then the cyclotomic integer (20) is zero iff $g_i(\zeta_{n/q})$ is zero for all $1 \leq i \leq t$.*

Proof. We shall show that $\zeta_n^{e_1}, \zeta_n^{e_2}, \cdots, \zeta_n^{e_t}$ are linearly independent over $\mathbb{Q}(\zeta_n^q) = \mathbb{Q}(\zeta_{n/q})$. For $1 \leq i \leq t$, set $s_i = 1 + (i-1)Tn/q$, where T is an integer that is congruent to $(n/q)^{-1} \pmod q$. Since for every i, $s_i \mod n/q = 1$ and s_i

mod $q = i < q$, so $gcd(s_i, n) = 1$ and $\sigma^{(s_i)} \in Gal(\mathbb{Q}(\zeta_n)/\mathbb{Q})$, which fixes $\mathbb{Q}(\zeta_{n/q})$. We only need to prove the matrix

$$\begin{aligned} V &= \begin{pmatrix} \sigma^{(s_1)}(\zeta_n^{e_1}) & \sigma^{(s_1)}(\zeta_n^{e_2}) & \cdots & \sigma^{(s_1)}(\zeta_n^{e_t}) \\ \sigma^{(s_2)}(\zeta_n^{e_1}) & \sigma^{(s_2)}(\zeta_n^{e_2}) & \cdots & \sigma^{(s_2)}(\zeta_n^{e_t}) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^{(s_t)}(\zeta_n^{e_1}) & \sigma^{(s_t)}(\zeta_n^{e_2}) & \cdots & \sigma^{(s_t)}(\zeta_n^{e_t}) \end{pmatrix} \\ &= \begin{pmatrix} \zeta_n^{e_1 s_1} & \zeta_n^{e_2 s_1} & \cdots & \zeta_n^{e_t s_1} \\ \zeta_n^{e_1 s_2} & \zeta_n^{e_2 s_2} & \cdots & \zeta_n^{e_t s_2} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_n^{e_1 s_t} & \zeta_n^{e_2 s_t} & \cdots & \zeta_n^{e_t s_t} \end{pmatrix} \end{aligned}$$

is nonsingular. In fact,

$$\begin{aligned} det(V) &= (\prod_{i=1}^{t} \zeta_n^{e_i}) \times \\ &\quad \begin{vmatrix} \zeta_n^{e_1(s_1-1)} & \zeta_n^{e_2(s_1-1)} & \cdots & \zeta_n^{e_t(s_1-1)} \\ \zeta_n^{e_1(s_2-1)} & \zeta_n^{e_2(s_2-1)} & \cdots & \zeta_n^{e_t(s_2-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_n^{e_1(s_t-1)} & \zeta_n^{e_2(s_t-1)} & \cdots & \zeta_n^{e_t(s_t-1)} \end{vmatrix} \\ &= (\prod_{i=1}^{t} \zeta_n^{e_i}) \times \\ &\quad \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \zeta_n^{\frac{e_1 Tn}{q}} & \zeta_n^{\frac{e_2 Tn}{q}} & \cdots & \zeta_n^{\frac{e_t Tn}{q}} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_n^{\frac{e_1(t-1)Tn}{q}} & \zeta_n^{\frac{e_2(t-1)Tn}{q}} & \cdots & \zeta_n^{\frac{e_t(t-1)Tn}{q}} \end{vmatrix}, \end{aligned}$$

where the matrix in the last line is Vandermonde. Hence

$$det(V) = \prod_{i=1}^{t} \zeta_n^{e_i} \prod_{1 \le i < j \le t} (\zeta_n^{e_j Tn/q} - \zeta_n^{e_i Tn/q}).$$

If $e_j \not\equiv e_i \pmod q$, then $e_j Tn/q \not\equiv e_i Tn/q \pmod n$. Hence $det(V) \neq 0$ and $\zeta_n^{e_1}, \zeta_n^{e_2}, \cdots, \zeta_n^{e_t}$ are linearly independent over $\mathbb{Q}(\zeta_{n/q})$ by Lemma 8.

□

Remark: The lemma implies that if n is free of prime factors less than $k+1$, then $f(\zeta_n)$ cannot be zero if the sparseness of f is k .

Lemma 10. *If $q^2|n$, then the cyclotomic integer (20) is zero iff $g_i(\zeta_{n/q})$ is zero for all $1 \le i \le t$.*

Proof. For $1 \le i \le t$, we define u_i to be $1 + (i-1)n/q$. Since for any prime dividing n, it must divide $(i-1)n/q$, we have that $gcd(u_i, n) = 1$. It is easy to see that $\sigma^{(u_i)}$ fixes $\mathbb{Q}(\zeta_{n/q})$. Just like what we do in the proof of Lemma 9, we compute the determinant of the matrix $W = (\sigma_n^{(u_i)}(\zeta_n^{e_j}))_{1 \le i,j \le t}$:

$$\begin{aligned}
& \begin{vmatrix} \zeta_n^{e_1 u_1} & \zeta_n^{e_2 u_1} & \cdots & \zeta_n^{e_t u_1} \\ \zeta_n^{e_1 u_2} & \zeta_n^{e_2 u_1} & \cdots & \zeta_n^{e_t u_2} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_n^{e_1 u_t} & \zeta_n^{e_2 u_t} & \cdots & \zeta_n^{e_t u_t} \end{vmatrix} \\
= \quad & (\prod_{i=1}^{t} \zeta_n^{e_i}) \times \\
& \begin{vmatrix} \zeta_n^{e_1(u_1-1)} & \zeta_n^{e_2(u_1-1)} & \cdots & \zeta_n^{e_t(u_1-1)} \\ \zeta_n^{e_1(u_2-1)} & \zeta_n^{e_2(u_2-1)} & \cdots & \zeta_n^{e_t(u_2-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_n^{e_1(u_t-1)} & \zeta_n^{e_2(u_t-1)} & \cdots & \zeta_n^{e_t(u_t-1)} \end{vmatrix} \\
= \quad & (\prod_{i=1}^{t} \zeta_n^{e_i}) \times \\
& \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \zeta_n^{\frac{e_1 n}{q}} & \zeta_n^{\frac{e_2 n}{q}} & \cdots & \zeta_n^{\frac{e_t n}{q}} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_n^{\frac{(t-1)e_1 n}{q}} & \zeta_n^{\frac{(t-1)e_2 n}{q}} & \cdots & \zeta_n^{\frac{(t-1)e_t n}{q}} \end{vmatrix},
\end{aligned}$$

where we need to compute the determinant of a Vandermonde matrix in the last line. Hence

$$det(W) = \prod_{i=1}^{t} \zeta_n^{e_i} \prod_{1 \le i < j \le t} (\zeta_n^{e_j n/q} - \zeta_n^{e_i n/q}).$$

If $e_j \not\equiv e_i \pmod{q}$, then $e_j n/q \not\equiv e_i n/q \pmod{n}$. Hence $det(W) \neq 0$ and $\zeta_n^{e_1}, \zeta_n^{e_2}, \cdots, \zeta_n^{e_t}$ are linearly independent over $\mathbb{Q}(\zeta_{n/q})$ by Lemma 8.

□

The remaining case is that $t = q$ is a prime and $gcd(q, n/q) = 1$. In this case, the q integers $n/q, 2n/q, \cdots, (q-1)n/q$ and n fall in different classes modulo q, so we can rewrite (20) in the form

$$\begin{aligned}
& \tilde{g}_1(\zeta_n^q)\zeta_n^{n/q} + \tilde{g}_2(\zeta_n^q)\zeta_n^{2n/q} + \cdots \\
& \quad + \tilde{g}_{q-1}(\zeta_n^q)\zeta_n^{(q-1)n/q} + \tilde{g}_q(\zeta_n^q)
\end{aligned} \tag{21}$$

Lemma 11. *If q is a prime and $gcd(q, n/q) = 1$, then the cyclotomic integer (21) is zero iff $\tilde{g}_1(\zeta_{n/q}) = \tilde{g}_2(\zeta_{n/q}) = \cdots = \tilde{g}_q(\zeta_{n/q})$.*

Proof. We have that

$$1 + \zeta_n^{n/q} + \zeta_n^{2n/q} + \cdots + \zeta_n^{(q-1)n/q} = 0.$$

Hence $1 = -\zeta_n^{n/q} - \zeta_n^{2n/q} - \cdots - \zeta_n^{(q-1)n/q}$. Substituting this into (21), we obtain

$$\begin{gathered}(\tilde{g}_1(\zeta_n^q) - \tilde{g}_q(\zeta_n^q))\zeta_n^{n/q} + \\ (\tilde{g}_2(\zeta_n^q) - \tilde{g}_q(\zeta_n^q))\zeta_n^{2n/q} + \\ \cdots + (\tilde{g}_{q-1}(\zeta_n^q) - \tilde{g}_q(\zeta_n^q))\zeta_n^{(q-1)n/q}. \end{gathered} \tag{22}$$

Lemma 9 implies that (22) is zero iff $\tilde{g}_i(\zeta_n^q) - \tilde{g}_q(\zeta_n^q) = 0$ for all $1 \le i \le q-1$. That is equivalent to

$$\tilde{g}_1(\zeta_{n/q}) = \tilde{g}_2(\zeta_{n/q}) = \cdots = \tilde{g}_q(\zeta_{n/q}).$$

□

3.2 Algorithm and time complexity analysis

Based on the lemmas in the previous section, we shall take a divide-and-conquer approach to design a zero testing algorithm for sparse cyclotomic integers. To guarantee polynomial time complexity, when Lemma 11 applies, we pick the $\tilde{g}_M(x)$ with fewest number of nonzero terms among all $\tilde{g}_i(x)$'s in (21), and test whether $\tilde{g}_i(\zeta_{n/q}) - \tilde{g}_M(\zeta_{n/q})$ equals to zero for all $i \neq M$, $1 \le i \le q$. The algorithm is described in Figure 3, whose inputs consist of an integral polynomial $f(x)$ given in sparse form and an integer n. The degree of f is less than n. The algorithm outputs "Yes" if $f(\zeta_n) = 0$. Otherwise it outputs "No".

Theorem 5. *The algorithm* **zerotesting(**$f(x)$**,** n**)** *runs in time* $O(k^3 \log n(\log n + \log m)^2)$*, where* k *is the sparseness of* $f(x)$ *and* m *is the height of* $f(x)$*.*

The following lemma is useful in proving the theorem.

Lemma 12. *Let* $t \ge 4$ *be a positive integer. Let* $a_1 \le a_2 \le \cdots \le a_t$ *be positive integers. Then*

1. $(\sum_{i=1}^t a_i)^2 > \sum_{i=1}^t a_i^2$;
2. $(\sum_{i=1}^t a_i)^2 > \sum_{i=2}^t (a_i + a_1)^2$;

1. If $f(x)$ is a zero polynomial, then return "Yes".

2. Let k be the sparseness of $f(x)$. Write $n = p_1^{\beta_1} p_2^{\beta_2} \cdots p_l^{\beta_l} r$, where $p_1, p_2, \cdots, p_l$ are primes less than $k+1$ and r does not have prime factors less than $k+1$.

3. If $r = 1$ and $l \leq 2$, then if

$$(x^n - 1) | f(x) \prod_{p|n} (x^{n/p} - 1),$$

return "yes", else return "no".

4. Let $q = \max\{p_1, p_2, \cdots, p_l, r\}$. Write $f(\zeta_n)$ as

$$g_t(\zeta_n^q)\zeta_n^{e_t} + \cdots + g_2(\zeta_n^q)\zeta_n^{e_2} + g_1(\zeta_n^q)\zeta_n^{e_1}$$

where $e_i \not\equiv e_j \pmod{q}$ for $1 \leq i < j \leq t$. If $q^2 | n$, or $t < q$, go to Step 6.

5. Rewrite $f(\zeta_n)$ in the form:

$$\begin{array}{l} \tilde{g}_1(\zeta_n^q)\zeta_n^{n/q} + \tilde{g}_2(\zeta_n^q)\zeta_n^{2n/q} + \cdots \\ \quad + \tilde{g}_{q-1}(\zeta_n^q)\zeta_n^{(q-1)n/q} + \tilde{g}_q(\zeta_n^q); \end{array}$$

Let $\tilde{g}_M(x)$ be the polynomial with minimum number of nonzero terms among all $\tilde{g}_i(x)$; Do

$$\begin{array}{rcl} g_1(x) & \leftarrow & \tilde{g}_1(x) - \tilde{g}_M(x) \\ & \cdots & \\ g_{M-1}(x) & \leftarrow & \tilde{g}_{M-1}(x) - \tilde{g}_M(x) \\ g_M(x) & \leftarrow & \tilde{g}_{M+1}(x) - \tilde{g}_M(x) \\ & \cdots & \\ g_{t-1}(x) & \leftarrow & \tilde{g}_t(x) - \tilde{g}_M(x) \\ t & \leftarrow & t - 1. \end{array}$$

6. If for all $1 \leq i \leq t$, **zerotesting**($g_i(x), n/q$) outputs "yes", then return "yes", else return "no".

Figure 3: Algorithm zerotesting($f(x)$, n)

Proof. The first inequality is trivial. For the second one, we have

$$\begin{aligned}
& (\sum_{i=1}^{t} a_i)^2 - \sum_{i=2}^{t} (a_i + a_1)^2 \\
= & \sum_{i=1}^{t} a_i^2 + \sum_{1 \le i < j \le t} 2a_i a_j \\
& - \sum_{i=2}^{t} a_i^2 - (t-1)a_1^2 - \sum_{i=2}^{t} 2a_1 a_i \\
= & \sum_{2 \le i < j \le t} 2a_i a_j - (t-2)a_1^2 \\
> & \ 0
\end{aligned}$$

□

Now we are ready to prove Theorem 5.

Proof. This algorithm is recursive. There are at most $\sum_{i=1}^{l} \beta_i + 1 \le \log n$ many recursive levels. For $1 \le i \le \sum_{i=1}^{l} \beta_i + 1$, let $[g_{i1}(x), g_{i2}(x), \cdots, g_{ih_i}(x)]$ be the list of polynomials that are inputs of **zerotesting** in the recursive level i. At the first level, there is only one polynomial $f(x)$. Namely, $h_1 = 1$ and $g_{1,1}(x) = f(x)$. We shall show that for $2 \le i \le \sum_{i=1}^{l} \beta_i + 1$,

$$\sum_{1 \le j \le h_i} sps^2(g_{ij}(x)) \le \sum_{1 \le j \le h_{i-1}} sps^2(g_{(i-1)j}(x)). \tag{23}$$

Without loss of generality consider the function call of **zerotesting**$(g_{(i-1)1}(x), n')$. Suppose that in Step 4, we write $g_{(i-1)1}(\zeta_{n'})$ as

$$g_1(\zeta_{n'/q})\zeta_{n'}^{e_1} + g_2(\zeta_{n'/q})\zeta_{n'}^{e_2} + \cdots + g_\tau(\zeta_{n'/q})\zeta_{n'}^{e_\tau}$$

and that g_j has sparseness a_j for $1 \le j \le \tau$. Then $sps(g_{(i-1)1}(x)) = \sum_{j=1}^{\tau} a_i$. Again without loss of generality, assume that $a_1 \le a_2 \le \cdots \le a_\tau$. Suppose that $g_{i1}(x), g_{i2}(x), \cdots, g_{it}(x)$ are handled in Step 6 of **zerotesting**$(g_{(i-1)1}(x), n')$. The sparseness of $g_{i1}(x), g_{i2}(x), \cdots, g_{it}(x)$ are either $a_1, a_2, \cdots, a_\tau$ respectively, or are at most $a_2 + a_1, a_3 + a_1, \cdots, a_\tau + a_1$ respectively. In both cases, we have

$$\sum_{1 \le j \le t} sps^2(g_{ij}(x)) \le sps^2(g_{(i-1)1}(x)).$$

Sum up for all $g_{(i-1)j}(x)$, $1 \le j \le h_{i-1}$, we prove (23).

At the first level, there is only one polynomial with sparseness k. The algorithm will have at most $\log n$ many levels and will never handle more than k^2 many sparse cyclotomic integers in any recursive level. Each cyclotomic integers will have sparseness no larger than k, and have height no larger than nm, because in the worst case the height can only be doubled as the algorithm goes down one level. Therefore the time complexity is $O(k^2 \log n \times k(\log(nm))^2) = O(k^3 \log n(\log n + \log m)^2)$. □

4 Cyclotomic array testing

As it was mentioned both algorithms can be expressed in terms of operations with sparse vectors of exponentially large dimension. For better understanding of this feature we introduce a more general problem *cyclotomic array test* (CAT for brevity).

Note that the primality of dimensions of the tensor factors in the decomposition (8) is not essential for the decomposition (13). Discarding this primality condition leads to a notion of a *cyclotomic array* that was introduced by Steinberger in [21].

We define *a cyclotomic array of type* $(n_1, \ldots, n_r; n_{r+1})$ as a vector from the subspace

$$X = \bigoplus_{k=1}^{r} V_{n_1} \otimes \cdots \otimes V_{n_{k-1}} \otimes \mathbb{Q}\hat{1} \otimes V_{n_{k+1}} \otimes \cdots \otimes V_{n_r} \otimes V_{n_{r+1}} \tag{24}$$

of the space $V = V_{n_1} \otimes \cdots \otimes V_{n_{r+1}}$. *The support set* $\operatorname{supp} f$ of a vector

$$f = \sum_{j_1,\ldots,j_{r+1}} f_{j_1,\ldots,j_{r+1}} e_{j_1} \otimes e_{j_2} \otimes \ldots \otimes e_{j_{r+1}} \in V$$

is a set of all $(j_1, \ldots, j_{r+1})$ such that $f_{j_1,\ldots,j_{r+1}} \neq 0$. As above, $sps(f)$ stands for a cardinality of support set (sparseness).

Comparing (24) and (13) shows that vanishing sums are cyclotomic arrays satisfying the following conditions on dimensions:

- n_i, $1 \leq i \leq r$, are pairwise distinct and form the set of all prime divisors of the total dimension $\dim V = \prod_{i=1}^{r+1} n_i$;
- if p is a prime divisor of n_{r+1} then $p = n_i$ for some i.

Due to Lemma 5 the kernel representation for cyclotomic arrays has the same form as for vanishing sums.

The CAT problem is stated as follows.

Input: integers $n_1, n_2, \ldots, n_r; n_{r+1}$ and a list containing m pairs (a_I, I) where a_I is integer and I is a $(r+1)$-tuple of indices (all integers are written in binary).

Output: "Yes" if the vector $\sum_J a_J e_J$ is a cyclotomic array and "No" otherwise.

The CAT problem is a natural generalization of the CT problem. It is worth mentioning that the CAT problem is related to the transportation problem and to the marginal distributions of multivariate probabilistic distributions. Cyclotomic arrays form a right kernel of a planar multiindex transportation problem [24]. A link to marginal distributions is based on the following fact. If two n-variate distributions p_1 and p_2 have the same $(n-1)$-variate marginal distributions then $p_1 - p_2$ is orthogonal to a space of cyclotomic arrays. Applications of the CAT algorithms to these problems are the subject of future research.

Now we are going to show that both algorithms can be modified to solve the CAT problem.

The matrix multiplication algorithm. The modification of the algorithm for the CAT problem starts from the step 5 (matrix computation) because the input of the CAT problem contains all data used in the final part of the matrix multiplication algorithm.

Note that the asymptotics of time complexity of the modified algorithm does not change. So, the running time of the algorithm is $O(sps^2(f)L^3)$, where $L = \log \max_{j,I}(n_j, a_I)$.

The recursive algorithm. The modification of the algorithm for the CAT problem decreases tensor dimension recursively. It corresponds to decreasing of the degree of a root of unity in the basic version of the algorithm represented in Figure 3.

To check that a vector f is a cyclotomic array of type $(;0)$ the algorithm checks that the component of f is zero (in this case there is only one component). This step corresponds to the Step 3 in the basic version.

To check that a vector f is a cyclotomic array of type $(n_1, \ldots, n_r; n_{r+1})$ the algorithm partitions indices $(j_1, \ldots, j_r, j_{r+1})$ from the support set according to the value of j_{r+1}. The vector f can be expressed as a sum

$$f = \sum_{j \in S} f_j, \quad \text{where } f_j = \sum_{J : j_{r+1} = j} f_J e_J \tag{25}$$

and S is the set of all possible values of j_{r+1} in the support of f. Then the algorithm checks that each f_j is a cyclotomic array of type $(n_1, \ldots, n_r; 0)$. This step corresponds to the Step 4 in the basic version when $q = r$.

To check that a vector f is a cyclotomic array of type $(n_1, \ldots, n_r; 0)$ the algorithm partitions indices $(j_1, \ldots, j_r)$ from the support set according to the value of j_r. The vector f can be expressed as a sum

$$f = \sum_{j \in S} f_j, \quad \text{where } f_j = \sum_{J : j_r = j} f_J e_J \tag{26}$$

and S is the set of all possible values of j_r in the support of f.

If $|S| < n_r$ then the algorithm checks that each f_j is a cyclotomic array of type $(n_1, \ldots, n_{r-1}; 0)$. This step also corresponds to the step 4 in Figure 3 when $t < q$.

Otherwise the algorithm finds a vector f_{j_0} with the smallest support and check that each vector $f_j - f_{j_0}$ is a cyclotomic array of type $(n_1, \ldots, n_{r-1}; 0)$. It corresponds to the Step 5 in Figure 3.

The correctness of the modified Step 4 follows from Lemma 7. To prove the correctness of the modified Step 5 we need one more lemma.

Lemma 13. *Let*

$$f = \sum_{s=0}^{n_1 - 1} e_s \otimes f_s \in X_n.$$

Then $e_s \otimes (f_s - f_{s'}) \in X_n$ *for all for all* s, s'.

Proof. Form a vector

$$f' = \hat{1} \otimes f_{s'} = \sum_{s=0}^{n_1-1} e_s \otimes f_{s'}.$$

By construction, $f_{s'} \in X_n$. The lemma follows by applying the argument of the proof of Lemma 7 to the vector $f - f'$. □

The size of recursion tree is estimated in the same way as in the proof of Theorem 5. To estimate the running time of the modified algorithm we note that on each recursive step the algorithm sorts indices and makes additions/substractions. Let k be the sparseness of f. The sorting takes a time $O(k \log k)$ and the addition/substraction takes a time $O(r + L)$, where $L = \log \max_{j,I}(n_j, a_I)$ is the bit length of integers involved and r is the depth of the recursion tree (each recursive step can double coefficients). The running time of a recursive step is thus $O(k \log k + k(r + L))$. Since there are no more than $O(k^2 r)$ recursive steps, the overall running time of the modified algorithm is upperbounded by $k^3 r(\log k + r + L)$.

Remark. One can reduce the CT problem to the CAT problem. This reduction is actually the part of the matrix multiplication algorithm (steps 1–4 in Fig. 2). Combining the reduction and the modified recursive algorithm for the problem CAT we get another efficient zero tesing algorithm. It's time complexity is $O(sps(f) \log^3 n + sps^3(f) \log n(\log sps(f) + \log n + \log H(f))$.

5 Torsion point problem

We prove Theorem 3 in this section. Recall that it claims TP $\in$ NP. The input of the TP problem is a list of multivariate polynomials $f_1, \ldots, f_k \in \mathbb{Z}[x_1, \ldots, x_d]$ in sparse representation and a list of positive integers $n_1, \ldots, n_d$. The output is "Yes" if the system of equations

$$f_1(x) = f_2(x) = \cdots = f_k(x) = x_1^{n_1} - 1 = \cdots = x_d^{n_d} - 1 = 0 \qquad (27)$$

has solutions in $\mathbb{C}^d$ and "No" otherwise.

We show that CT $\in$ P implies TP $\in$ NP and thus Theorem 3 follows from Theorem 1.

The argument is straightforward. Coordinates of a solution of the system (27) can be expressed as powers of a primitive root of the degree $n = \prod_{j=1}^{d} n_d$. Note that $\log n$ is less than the input size. So, there is a short certificate $(\alpha_1, \ldots, \alpha_d)$ for a solution $x_j = \zeta_n^{\alpha_j}$ of a instance of the problem TP. To verify the certificate we express the values of the polynomials $f_1, \ldots, f_k$ at the point $(\zeta_n^{\alpha_1}, \ldots, \zeta_n^{\alpha_d})$ as cyclotomic integers and perform zero testing for each cyclotomic integer. It is easy to see that substitutions of $\zeta_n^{\alpha_j}$ and grouping terms according to powers of ζ_n can be done in polynomial time.

6 Concluding remarks

In this paper, we study the zero testing problem of sparse cyclotomic integers and some related problems. We present two deterministic polynomial time algorithms for the zero testing problem of sparse cyclotomic integers.

An immediate generalization of this problem: check whether

$$f(\zeta_n^{e_1}, \zeta_n^{e_2}, \cdots, \zeta_n^{e_k}) = 0$$

for a multivariate polynomial $f(x_1, x_2, \cdots, x_k)$ given by a formula and the exponents $e_1, e_2, \ldots, e_k$ and the degree n are represented in binary. For this problem the randomized argument outlined in Subsection 1.1 works well. So, the problem is in the class BPP. But the formula representation is more succinct than the sparse representation, which makes the derandomization harder. It is open whether this problem is in P.

Note also that if we use the circuit (straight-line program) representation of the polynomials then the randomized argument fails. In this case coefficients of polynomials can be doubly exponentially large. The complexity of this circuit zero testing is upperbounded by a finite level of counting hierarchy [1] (in partiular, the problem is in PSPACE). Lower complexity bounds for this problem are unknown.

Another interesting open problem is to decide whether $\sum_{j\in J} a_j \zeta_n^j$ is positive or negative if it is known to be real. This sign determination problem of sparse real cyclotomic integers appears to be much harder than the zero testing problem. It is related to the sum of square roots problem [11, 8], a famous open problem in computational geometry, which asks to determine the sign of

$$\sqrt{a_1} + \cdots + \sqrt{a_k} - \sqrt{b_1} - \cdots - \sqrt{b_k} \tag{28}$$

where a_i and b_i are positive integers. For a prime p, the square of the principle Gaussian sum $\sum_{i=1}^{p-1} (\frac{i}{p}) \zeta_p^i$ is p or $-p$. Hence $\sqrt{p} \in \mathbb{Q}(\zeta_{4p})$. Here $(\frac{i}{p})$ is the Legendre symbol:

$$(\frac{i}{p}) = \begin{cases} 1, & \text{if } i \text{ is a quadratic residue modulo } p, \\ -1, & \text{if } i \text{ is a quadratic nonresidue modulo } p. \end{cases}$$

Applying this observation, we can write (28) as sum of at most $\sum_{i=1}^k a_i + \sum_{i=1}^k b_i$ many t-th roots of unity, where $t \mid (4 \prod_{i=1}^k a_i \prod_{i=1}^k b_i)$. As a result, we obtain a *sparse* cyclotomic integer, assuming that a_i's and b_i's are bounded from above by a polynomial function on k. This means that we reduce the problem of comparing sums of square roots to the sign determination problem of sparse cyclotomic integers when a_i's and b_i's are small. Note that it is still open whether the former problem is in NP or not, even in the case when a_i's and b_i's are bounded from above by a polynomial function on k.

Acknowledgments

We are thankful to the unknown referees for their comments on the first version of the paper.

References

[1] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen and P. Bro Miltersen. On the Complexity of Numerical Analysis. IEEE Conference on Computational Complexity (2006), 331–339.

[2] J. Blomer. A probabilistic zero-test for expressions involving roots of rational numbers. In *Proc. ESA*, volume 1461 of *LNCS*, pages 151–162, 1998.

[3] E. Bach and J. Shallit. Algorithmic Number Theory, Vol. I: Efficient Algorithms, MIT Press, Cambridge, MA, 1996.

[4] N. G. de Bruijn. On the factorization of cyclic groups, Indag. Math. **15** (1953) 370–377.

[5] Zhi-Zhong Chen and Ming-Yang Kao. Reducing randomness via irrational numbers. *SIAM J. Comput.*, 29(4):1247–1256, 2000.

[6] Qi Cheng. Derandomization of Sparse Cyclotomic Integer Zero Testing, FOCS, 2007.

[7] J. H. Conway and A. J. Jones. Trigonometric Diophantine equations (On vanishing sums of roots of unity), Acta Arith. **30** (1976) 229–240

[8] E. D. Demaine, J. S. B. Mitchell, and J. O'Rourke. The open problems project: Problem 33. http://maven.smith.edu/~orourke/TOPP/.

[9] M. Filaseta, A. Granville, and A. Schinzel. Irreducibility and greatest common divisor algorithms for sparse polynomials. `http://www.math.sc.edu//~filaseta/papers/SparsePaper.pdf`

[10] M. Filaseta, A. Schinzel. On testing the divisibility of lacunary polynomials by cyclotomic polynomials, Math. Comp. **73** (2004) 957–965

[11] M. Garey, R.L. Graham, and D.S. Johnson. Some NP-complete geometric problems. In *Proc. ACM Symp. Theory Comp.*, pages 10–21, 1976.

[12] P. Koiran. Hilbert's Nullstellensatz is in the Polynomial Hierarchy. Journal of Complexity **12** (1996), no. 4, pp. 273–286.

[13] T. Y. Lam, K. H. Leung. On vanishing sums of roots of unity. J. Algebra **224** (2000) 91–109

[14] M. Mignotte. Identification of algebraic numbers. J. Algorithms, **3** (1982) 197–204

[15] D. A. Plaisted. New NP-hard and NP-complete polynomial and integer divisibility problems. Theoretical Computer Science **31** (1984) 125–138

[16] L. Rédei. Ein Beitrag zum Problem der Faktorisation von Abelschen Gruppen, Acta Math. Acad. Sci. Hungar. **1** (1950) 197–207

[17] L. Rédei. Natürliche Basen des Kreisteilungskörpers, Teil I. Abh. Math. Sem. Hamburg **23** (1959) 180–200

[18] J.M. Rojas. Efficiently Detecting Subtori and Torsion Points, proceedings of MAGIC 2005 (Midwest Algebra, Geometry, and their Interactions Conference, Oct. 7-11, 2005, Notre Dame University, Indiana), edited by A. Corso, J. Migliore, and C. Polini), pp. 213-233, Contemporary Mathematics, vol. 448, AMS Press, 2007.

[19] I. J. Schoenberg. A note on the cyclotomic polynomial, Mathematika **11** (1964) 131–136

[20] J.N.Schwartz. Fast probabilistic algorithms for verification of polynomial identities. J. of the ACM 27(1980), 701–717.

[21] J. P. Steinberger. Minimal vanishing sums of roots of unity with large coefficients. `http://www.math.ucdavis.edu/~jpsteinb/vanishing.ps`

[22] T. Tao. An uncertainty principle for cyclic groups of prime order. *Math. Res. Lett.*, 12(1):121–127, 2005.

[23] S.Tarasov, M. Vyalyi. An efficient algorithm for zero-testing of a lacunary polynomial at the roots of unity. In *Proc. CSR2007*, LNCS Vol. 4649, 2007. P. 397–406.

[24] E. Titova, V. Shevchenko. Left and right kernels of a planar multiindex transportation problem. Proc. of VIII International seminar 'Discrete mathematics and applications'. MSU, Moscow (2004) 229–230. (In Russian)