THE OPEN BOOK SERIES 1

ANTS X
Proceedings of the Tenth
Algorithmic Number Theory Symposium

Counting value sets: algorithm and complexity

Qi Cheng, Joshua E. Hill, and Daqing Wan

msp

THE OPEN BOOK SERIES **1** (2013)
Tenth Algorithmic Number Theory Symposium
dx.doi.org/10.2140/obs.2013.1.235

Counting value sets: algorithm and complexity

Qi Cheng, Joshua E. Hill, and Daqing Wan

Let p be a prime. Given a polynomial in $\mathbb{F}_{p^m}[x]$ of degree d over the finite field $\mathbb{F}_{p^m}$, one can view it as a map from $\mathbb{F}_{p^m}$ to $\mathbb{F}_{p^m}$, and examine the image of this map, also known as the *value set* of the polynomial. In this paper, we present the first nontrivial algorithm and the first complexity result on explicitly computing the cardinality of this value set. We show an elementary connection between this cardinality and the number of points on a family of varieties in affine space. We then apply Lauder and Wan's p-adic point-counting algorithm to count these points, resulting in a nontrivial algorithm for calculating the cardinality of the value set. The running time of our algorithm is $(pmd)^{O(d)}$. In particular, this is a polynomial-time algorithm for fixed d if p is reasonably small. We also show that the problem is #P-hard when the polynomial is given in a sparse representation, $p = 2$, and m is allowed to vary, or when the polynomial is given as a straight-line program, $m = 1$ and p is allowed to vary. Additionally, we prove that it is NP-hard to decide whether a polynomial represented by a straight-line program has a root in a prime-order finite field, thus resolving an open problem proposed by Kaltofen and Koiran.

1. Introduction

Let $f \in \mathbb{F}_q[x]$ be a polynomial of degree d with coefficients in a finite field having $q = p^m$ elements, where p is prime. Denote the image set of this polynomial by

$$V_f = \left\{ f(\alpha) \mid \alpha \in \mathbb{F}_q \right\}$$

and denote the cardinality of this set by $\#(V_f)$.

There are a few trivial bounds on $\#(V_f)$ that can be immediately established. There are only q elements in the field, so $\#(V_f) \leq q$. Additionally, any polynomial

MSC2010: primary 11Y16; secondary 11Y40, 68Q17.
Keywords: finite field, polynomial value set cardinality, point counting, polynomial time, randomized polynomial time, RP-reduction, NP-hard, #P-hard, straight-line program, sparse polynomial, subset sum problem.

of degree d can have at most d roots, thus for all $a \in V_f$, $f(x) = a$ is satisfied at most d times. This is true for every element in V_f, so $\#(V_f)d \geq q$, whence

$$\left\lceil \frac{q}{d} \right\rceil \leq \#(V_f) \leq q,$$

where $\lceil \cdot \rceil$ is the ceiling function.

Both of these bounds can be achieved: If $\#(V_f) = q$, then f is called a *permutation polynomial*, and if $\#(V_f) = \lceil q/d \rceil$, then f is said to have a *minimal value set*.

The problem of computing $\#(V_f)$ has been studied in various forms for at least the last 115 years, but exact formulas for $\#(V_f)$ are known only for polynomials of very specific forms. Results that apply to general polynomials are asymptotic in nature, or provide estimates whose errors have reasonable bounds only on average [14].

The fundamental problem of determining the value set cardinality $\#(V_f)$ can be thought of as a much more general version of the problem of determining whether a particular polynomial is a permutation polynomial. Shparlinski [17] provides a baby-step giant-step type test that determines if a given polynomial is a permutation polynomial by extending the ideas in [20] to an algorithm that runs in time $\tilde{O}((dq)^{6/7})$. This is still fully exponential in $\log q$. Ma and von zur Gathen [13] provide a ZPP (zero-error probabilistic polynomial-time) algorithm for testing if a given polynomial is a permutation polynomial. According to [10], the first deterministic polynomial-time algorithm for testing permutation polynomials was obtained by Lenstra using the classification of exceptional polynomials, which in turn depends on the classification of finite simple groups. Subsequently, an elementary approach based on the Gao-Kaltofen-Lauder factorization algorithm was given by Kayal [10].

Essentially nothing is known about the complexity of the more general problem of exactly computing $\#(V_f)$, and no nontrivial algorithms for this problem are known. For instance, no baby-step giant-step type algorithm for computing $\#(V_f)$ is known, and no probabilistic polynomial-time algorithm for this problem is known. Finding a nontrivial algorithm and proving a nontrivial complexity result for the value counting problem were raised as open problems in [13], where a probabilistic approximation algorithm is given. In this paper, we provide the first nontrivial algorithm and the first nontrivial complexity result for the exact counting of the value set problem.

1A. ***Our results.*** Perhaps the most obvious method to calculate $\#(V_f)$ is to evaluate the polynomial at each point in $\mathbb{F}_q$ and count how many distinct images result. This algorithm has a time and space complexity $(dq)^{O(1)}$. One can also approach this problem by operating on points in the codomain. One has $f(x) = a$ for some

$x \in \mathbb{F}_q$ if and only if $f_a(X) = f(X) - a$ has a zero in $\mathbb{F}_q$; this algorithm again has a time complexity $(dq)^{O(1)}$, but the space complexity is improved considerably to $(d \log q)^{O(1)}$.

In this paper we present several results on determining the cardinality of value sets. On the algorithmic side, we show an elementary connection between this cardinality and the number of points on a family of varieties in affine space. We then apply Lauder and Wan's p-adic point-counting algorithm [12], resulting in a nontrivial algorithm for calculating the image set cardinality in the case that p is sufficiently small (that is, $p = O((d \log q)^C)$ for some positive constant C). Precisely, we have the following.

Theorem 5.2. *There exists an explicit deterministic algorithm and an explicit polynomial R such that for any $f \in \mathbb{F}_q[x]$ of degree d, where $q = p^m$ (p prime), the algorithm computes $\#(V_f)$, the cardinality of the image set, in a number of bit operations bounded by $R(m^d d^d p^d)$.*

The running time of this algorithm is polynomial in both p and m, but is exponential in d. In particular, this is a polynomial-time algorithm for fixed d if the characteristic p is small: $q = p^m$ can be large, but $p = O((d \log q)^C)$.

On the complexity side, we have several hardness results on the value set problem. We frame these results using some standard classes in complexity theory, which we outline here. NP is the complexity class of decision problem whose positive solutions can be verified in polynomial time. NP-hard is the computational class of decision problems that all NP problems can be reduced to using a polynomial-time reduction. NP-complete is the complexity class of all NP-hard problems whose solution can be verified in polynomial time (that is, NP-complete is the intersection of NP-hard and NP). Co-NP-complete is the complexity class of problems where answering the logical complement of the decision problem is NP-complete.

The corresponding counting complexity theory classes that we use are as follows. #P (read "sharp-P") is the set of counting problems whose corresponding decision problem is in NP. #P-hard is the computational class of counting problems that all #P problems can be reduced to using a polynomial-time counting reduction. #P-complete is the intersection of #P-hard and #P.

With a field of characteristic 2, we have the following.

Theorem 4.3. *The problem of counting the value set of a sparse polynomial over a finite field of characteristic 2 is #P-hard.*

The central approach in our proof of this theorem is to reduce the problem of counting satisfying assignments for a 3SAT formula to the problem of value set counting.

Over a prime-order finite field, we have the following.

Theorem 4.6. *Over a prime-order finite field $\mathbb{F}_p$, the problem of counting the value set is #P-hard under RP-reduction* (*randomized polynomial-time reduction*) *if the polynomial is given as a straight-line program.*

Additionally, we prove that it is NP-hard to decide whether a polynomial in $\mathbb{Z}[x]$ represented by a straight-line program has a root in a prime-order finite field, thus resolving an open problem proposed in [7; 8]. We accomplish the complexity results over prime-order finite fields by reducing the prime-order finite field subset sum problem (PFFSSP) to these problems.

In the PFFSSP, given a prime p, an integer b, and a set of integers $S = \{a_1, a_2, \dots, a_t\}$, we want to decide the solvability of the equation

$$a_1x_1 + a_2x_2 + \cdots + a_tx_t \equiv b \pmod{p}$$

with $x_i \in \{0, 1\}$ for $1 \le i \le t$. The main idea comes from the observation that if $t < \log p/3$, there is a sparse polynomial $\alpha(x) \in \mathbb{F}_p[x]$ such that as x runs over $\mathbb{F}_p$, the vector

$$\big(\alpha(x), \alpha(x+1), \dots, \alpha(x+t-1)\big)$$

runs over all the elements in $\{0, 1\}^t$. In fact, a lightly modified version of the quadratic character $\alpha(x) = (x^{(p-1)/2} + x^{p-1})/2$ suffices. So the PFFSSP can be reduced to deciding whether the sparse shift polynomial $\sum_{i=0}^{t-1} a_{i+1}\alpha(x+i) - b = 0$ has a solution in $\mathbb{F}_p$.

2. Background

2A. ***The subset sum problem.*** To prove the complexity results, we use the subset sum problem (SSP) extensively. The SSP is a well-known problem in computer science; we describe three versions of it. Let an integer b and a set of positive integers $S = \{a_1, a_2, \dots, a_t\}$ be given.

(1) Decision version: The goal is to decide whether there exists a subset $T \subseteq S$ such that the sum of all the integers in T equals b.

(2) Search version: The goal is to find a subset $T \subseteq S$ such that the sum of all the integers in T equals b.

(3) Counting version: The goal is to count the number of subsets $T \subseteq S$ such that the sum of all the integers in T equals b.

The decision version of the SSP is a classical NP-complete problem. The counting version of the SSP is #P-complete, which can be easily derived from proofs of the NP-completeness of the decision version, for example [5, Theorem 34.15].

One can view the SSP as a problem of solving the linear equation

$$a_1x_1 + a_2x_2 + \cdots + a_tx_t = b$$

with $x_i \in \{0, 1\}$ for $1 \le i \le t$. The prime-order finite field subset sum problem is a similar problem where in addition to b and S, one is given a prime p, and the goal is to decide the solvability of the equation

$$a_1 x_1 + a_2 x_2 + \cdots + a_t x_t \equiv b \pmod p$$

with $x_i \in \{0, 1\}$ for $1 \le i \le t$.

Proposition 2.1. *The prime-order finite field subset sum problem is NP-hard under RP-reduction.*

Proof. To reduce the subset sum problem to the prime-order finite field subset sum problem, one finds a prime $p > \sum_{i=1}^{t} a_i$, which can be done in randomized polynomial time. □

Remark. To make the reduction deterministic, one needs to derandomize the problem of finding a large prime, which appears to be difficult [18].

2B. *Polynomial representations.* There are different ways to represent a polynomial over a field $\mathbb{F}$. The dense representation lists all the coefficients of a polynomial, including the zero coefficients. The sparse representation lists only the nonzero coefficients, along with the degrees of the corresponding terms. If most of the coefficients of a polynomial are zero, then the sparse representation is much shorter than the dense representation. A sparse shift representation of a polynomial in $\mathbb{F}[x]$ is a list of n triples $(a_i, b_i, e_i) \in \mathbb{F} \times \mathbb{F} \times \mathbb{Z}_{\ge 0}$ which represents the polynomial

$$\sum_{1 \le i \le n} a_i (x + b_i)^{e_i}.$$

More generally, a straight-line program for a univariate polynomial in $\mathbb{Z}[x]$ or $\mathbb{F}_p[x]$ is a sequence of assignments, starting from $x_1 = 1$ and $x_2 = x$. After that, the i-th assignment has the form

$$x_i = x_j \odot x_k$$

where $0 \le j, k < i$ and $\odot$ is one of the three operations $+$, $-$, $\times$. We first let α be an element in $\mathbb{F}_{p^m}$ such that $\mathbb{F}_{p^m} = \mathbb{F}_p[\alpha]$. A straight-line program for a univariate polynomial in $\mathbb{F}_{p^m}[x]$ can be defined similarly, except that the sequence starts from $x_1 = \alpha$ and $x_2 = x$. One can verify that a straight-line program computes a univariate polynomial, and that sparse polynomials and sparse shift polynomials have short straight-line programs. A polynomial produced by a short straight-line program may have very high degree, and most of its coefficients may be nonzero, so it may be costly to write it in either a dense form or a sparse form.

3. Hardness of solving straight-line polynomials

It is known that deciding whether there is a root in a finite field for a sparse polynomial is NP-hard [11]. In a related work, it was shown that deciding whether there is a p-adic rational root for a sparse polynomial is NP-hard [1]. However, the complexity of deciding the solvability of a straight-line polynomial in $\mathbb{Z}[x]$ within a prime-order finite field was not known. This open problem was proposed in [7] and [8]. We resolve this problem within this section, and this same idea will be used later on to prove the hardness result of the value set counting problem.

Let p be an odd prime. Let χ be the quadratic character modulo p; that is, $\chi(x)$ equals 1, -1, or 0, depending on whether x is a quadratic residue, a quadratic nonresidue, or is congruent to 0 modulo p. For $x \in \mathbb{F}_p$, we have $\chi(x) = x^{(p-1)/2}$. Consider the list

$$\chi(1), \chi(2), \ldots, \chi(p-1). \tag{1}$$

It is a sequence in $\{1, -1\}^{p-1}$. The following bound is a standard consequence of the celebrated Weil bound for character sums; see [16] for a detailed proof.

Proposition 3.1. *Let $(b_1, b_2, \ldots, b_t)$ be a sequence in $\{1, -1\}^t$. Then the number of $x \in \mathbb{F}_p$ such that*

$$\chi(x) = b_1, \chi(x+1) = b_2, \ldots, \chi(x+t-1) = b_t$$

lies between $p/2^t - t(3 + \sqrt{p})$ and $p/2^t + t(3 + \sqrt{p})$.

The proposition implies that if $t < (\log p)/3$, then every possible sequence in $\{-1, 1\}^t$ occurs as a consecutive subsequence in expression (1). In many situations it is more convenient to use binary 0/1 sequences, which suggests instead using the polynomial $(x^{(p-1)/2} + 1)/2$, but this results in a small problem at $x = 0$. We instead use the sparse polynomial

$$\alpha(x) = (x^{(p-1)/2} + x^{p-1})/2. \tag{2}$$

The polynomial $\alpha(x)$ takes values in $\{0, 1\}$ if $x \in \mathbb{F}_p$, and $\alpha(x) = 1$ if and only if $\chi(x) = 1$.

Corollary 3.2. *If $t < (\log p)/3$, then for any binary sequence $(b_1, b_2, \ldots, b_t) \in \{0, 1\}^t$ there exists an $x \in \mathbb{F}_p$ such that*

$$\alpha(x) = b_1,\ \alpha(x+1) = b_2, \ldots, \alpha(x+t-1) = b_t.$$

In other words, if $t < (\log p)/3$, the map

$$x \mapsto \big(\alpha(x), \alpha(x+1), \ldots, \alpha(x+t-1)\big)$$

is a *surjective* map from $\mathbb{F}_p$ to $\{0, 1\}^t$; one can view this map as sending an algebraic object to a combinatorial object.

Given a straight-line polynomial $f(x) \in \mathbb{Z}[x]$ and a prime p, how hard is it to decide whether the polynomial has a solution in $\mathbb{F}_p$? We now prove that this problem is NP-hard.

Theorem 3.3. *Given a sparse shift polynomial $f(x) \in \mathbb{Z}[x]$ and a large prime p, it is NP-hard to decide whether $f(x)$ has a root in $\mathbb{F}_p$ under RP-reduction.*

Proof. We reduce the (decision version of the) subset sum problem to this problem. Given $b \in \mathbb{Z}_{\geq 0}$ and $S = \{a_1, a_2, \dots, a_t\} \subseteq \mathbb{Z}_{\geq 0}$, one can find a prime p such that $p > \max(2^{3t}, \sum_{i=1}^{t} a_i)$ and construct a sparse shift polynomial

$$\beta(x) = \sum_{i=0}^{t-1} a_i \alpha(x+i) - b. \tag{3}$$

If the polynomial has a solution modulo p, then the answer to the subset sum problem is "yes", since for every $x \in \mathbb{F}_p$ we have $\alpha(x+i) \in \{0, 1\}$.

In the other direction, if the answer to the subset sum problem is "yes", then according to Corollary 3.2, the polynomial has a solution in $\mathbb{F}_p$. Note that the reduction can be computed in randomized polynomial time. □

4. Complexity of the value set counting problem

In this section, we prove several results about the complexity of the value set counting problem.

4A. ***Finite fields of characteristic 2.*** We will use a problem about NC_5^0 circuits to prove that counting the value set of a sparse polynomial in a field of characteristic 2 is #P-hard. A Boolean circuit is in NC_5^0 if every output bit of the circuit depends only on at most 5 input bits. We can view a circuit with n input bits and m output bits as a map from $\{0, 1\}^n$ to $\{0, 1\}^m$ and call the image of the map the *value set* of the circuit. The following proposition is implied in [6]; we provide a sketch of the proof.

Proposition 4.1. *Given a* 3SAT *formula with n variables and m clauses, one can construct in polynomial time an NC_5^0 circuit with $n+m$ input bits and $n+m$ outputs bits, such that if there are M satisfying assignments for the* 3SAT *formula, then the cardinality of the value set of the NC_5^0 circuit is $2^{n+m} - 2^{m-1}M$. In particular, if the* 3SAT *formula can not be satisfied, then the circuit computes a permutation from $\{0, 1\}^{n+m}$ to $\{0, 1\}^{n+m}$.*

Proof. Denote the variables and the clauses of the 3SAT formula by $x_1, x_2, \dots, x_n$ and $C_1, C_2, \dots, C_m$, respectively. Build a circuit with $n+m$ input bits and $n+m$ output bits as follows. The input bits will be denoted by $x_1, x_2, \dots, x_n$

and $y_1, y_2, \dots, y_m$, and the output bits will be denoted by $z_1, z_2, \dots, z_n$ and $w_1, w_2, \dots, w_m$. Set $z_i = x_i$ for $1 \le i \le n$, and set

$$w_i = \bigl(C_i \wedge (y_i \oplus y_{(i+1 \,(\mathrm{mod}\, m))})\bigr) \vee (\neg C_i \wedge y_i)$$

for $1 \le i \le m$. In other words, if C_i is evaluated to be TRUE, then output $y_i \oplus y_{(i+1 \,(\mathrm{mod}\, m))}$ as w_i, and otherwise output y_i as w_i. Note that C_i depends only on 3 variables from $\{x_1, x_2, \dots, x_n\}$, so we obtain an NC_5^0 circuit. After fixing an assignment to the x_i, the z_i are also fixed, and the transformation from $(y_1, y_2, \dots, y_m)$ to $(w_1, w_2, \dots, w_m)$ is linear over $\mathbb{F}_2$. One can verify that the linear transformation has rank $m-1$ if the assignment satisfies all the clauses, and it has rank m (that is, it has full rank) if some of the clauses are not satisfied. So the cardinality of the value set of the circuit is

$$M 2^{m-1} + (2^n - M) 2^m = 2^{n+m} - 2^{m-1} M. \qquad \square$$

If we replace the Boolean gates in the NC_5^0 circuit by algebraic gates over $\mathbb{F}_2$, we obtain an algebraic circuit that computes a polynomial map from $\mathbb{F}_2^{n+m}$ to itself, where each polynomial depends only on 5 variables and has degree equal to or less than 5. There is an $\mathbb{F}_2$-basis for $\mathbb{F}_{2^{n+m}}$, say $\omega_1, \omega_2, \dots, \omega_{n+m}$, which induces a bijection from $\mathbb{F}_2^{n+m}$ to $\mathbb{F}_{2^{n+m}}$ given by

$$(x_1, x_2, \dots, x_{n+m}) \mapsto x = \sum_{i=1}^{n+m} x_i \omega_i;$$

the inverse of this map can be represented by sparse polynomials in $\mathbb{F}_{2^{n+m}}[x]$. Using this fact, we can replace the input bits of the algebraic circuit by sparse polynomials, and collect the output bits together using the base to form a single element in $\mathbb{F}_{2^{n+m}}$. We thus obtain a sparse univariate polynomial in $\mathbb{F}_{2^{n+m}}[x]$ from the NC_5^0 circuit such that their value sets have the same cardinality. We thus have the following theorem.

Theorem 4.2. *Given a* 3SAT *formula with n variables and m clauses, one can construct in polynomial time a sparse polynomial $\gamma(x)$ over $\mathbb{F}_{2^{n+m}}$ such that the value set of $\gamma(x)$ has cardinality $2^{n+m} - 2^{m-1} M$, where M is the number of satisfying assignments of the* 3SAT *formula.*

Since counting the number of satisfying assignments for a 3SAT formula is known to be #P-complete, we have our main theorem.

Theorem 4.3. *The problem of counting the value set of a sparse polynomial over a finite field of characteristic* 2 *is #P-hard.*

Corollary 4.4. *The set of sparse permutation polynomials over finite fields of characteristic* 2 *is co-NP-complete.*

4B. ***Prime-order finite fields.*** The construction in Theorem 4.2 relies on building extensions over $\mathbb{F}_2$. The technique cannot be adopted easily to the prime-order finite field case. We will prove that counting the value set of a straight-line polynomial over a prime-order finite field is #P-hard. We reduce the counting version of the subset sum problem to the value set counting problem.

Theorem 4.5. *Given access to an oracle that solves the value set counting problem for straight-line polynomials over prime-order finite fields, there is a randomized polynomial-time algorithm solving the counting version of the* SSP.

Proof. Suppose we are given an instance of the counting subset sum problem, say b with the set $S = \{a_1, a_2, \dots, a_n\}$. If $b > \sum_{i=1}^{n} a_i$ we answer 0, while if $b = 0$ we answer 1. Otherwise, we find a prime $p > \max(2^{3t}, 2\sum_{i=1}^{n} a_i)$ and ask the oracle to count the value set of the sparse shift polynomial

$$f(x) := \bigl(1 - \beta(x)^{p-1}\bigr)\Biggl(\sum_{i=0}^{t-1} \alpha(x+i)2^i\Biggr)$$

over the prime-order field $\mathbb{F}_p$, where $\alpha(x)$ and $\beta(x)$ are as defined in (2) and (3), respectively. We output the answer $\#(V_f) - 1$, which is easily seen to be exactly the number of subsets of $\{a_1, \dots, a_n\}$ that sum to b. □

Since the counting version of the SSP is #P-complete, this theorem yields the following.

Theorem 4.6. *Over a prime-order finite field $\mathbb{F}_p$, the problem of counting the value set is #P-hard under RP-reduction, if the polynomial is given as a straight-line program.*

5. The image set and point counting

Proposition 5.1. *If $f \in \mathbb{F}_q[x]$ is a polynomial of degree $d > 0$, then the cardinality of its image set is*

$$\#(V_f) = \sum_{i=1}^{d} (-1)^{i-1} N_i \sigma_i\Bigl(1, \frac{1}{2}, \dots, \frac{1}{d}\Bigr) \tag{4}$$

where $N_k = \#(\{(x_1, \dots, x_k) \in \mathbb{F}_q^k \mid f(x_1) = \cdots = f(x_k)\})$ and σ_i denotes the i-th elementary symmetric function on d elements.

Proof. For any $y \in V_f$, define

$$\tilde{N}_{k,y} = \bigl\{(x_1, \dots, x_k) \in \mathbb{F}_q^k \mid f(x_1) = \cdots = f(x_k) = y\bigr\}$$

and denote the cardinalities of $\tilde{N}_{k,y}$ by $N_{k,y}$. We then see that

$$N_k = \sum_{y \in V_f} N_{k,y}. \tag{5}$$

Let us refer to the right-hand side of (4) as η; plugging (5) into this expression and rearranging, we get

$$\eta = \sum_{y \in V_f} \sum_{i=1}^{d} (-1)^{i-1} N_{i,y}\, \sigma_i\left(1, \frac{1}{2}, \dots, \frac{1}{d}\right).$$

Let us call the inner sum ω_y; that is,

$$\omega_y = \sum_{i=1}^{d} (-1)^{i-1} N_{i,y}\, \sigma_i\left(1, \frac{1}{2}, \dots, \frac{1}{d}\right).$$

If we can show that for all $y \in V_f$ we have $\omega_y = 1$, then we clearly have $\eta = \#(V_f)$.

Let $y \in V_f$ be fixed. Let $k = \#(f^{-1}(y))$. It is clear that $1 \le k \le d$ and $N_{i,y} = k^i$ for $0 \le i \le d$. Substituting this in, our expression mercifully becomes somewhat nicer:

$$\begin{aligned}
\omega_y &= 1 - \sum_{i=0}^{d} (-1)^i k^i \sigma_i\left(1, \frac{1}{2}, \dots, \frac{1}{d}\right) \\
&= 1 - \sum_{i=0}^{d} (-1)^i \sigma_i\left(k, \frac{k}{2}, \dots, \frac{k}{d}\right) \qquad (6) \\
&= 1 - \left[\left(1-k\right)\left(1-\frac{k}{2}\right)\cdots\left(1-\frac{k}{d}\right)\right] \qquad (7) \\
&= 1.
\end{aligned}$$

From step (6) to step (7), we are using the identity

$$\prod_{j=1}^{n} \left(\lambda - X_j\right) = \sum_{j=0}^{n} (-1)^j \lambda^{n-j} \sigma_j\left(X_1, \dots, X_n\right).$$

Note that the bracketed term of (7) is 0, as k must be an integer such that $1 \le k \le d$, so one term in the product will be 0. Thus, we have $\eta = \#(V_f)$, as desired. □

Proposition 5.1 gives us a way to express $\#(V_f)$ in terms of the numbers of rational points on a sequence of curves over $\mathbb{F}_q$. If we had a way of getting N_k for $1 \le k \le d$, then it would be easy to calculate $\#(V_f)$.

We proceed by examining a family of related spaces,

$$\tilde{N}_k = \left\{(x_1, \dots, x_k) \in \mathbb{F}_q^k \mid f(x_1) = \cdots = f(x_k)\right\}.$$

We immediately note that $N_k = \#(\tilde{N}_k)$.

Spaces similar to our $\tilde{N}_k$ have been used several times [19; 2] to establish various asymptotic results for $\#(V_f)$. The spaces used in these earlier papers require that $x_i \neq x_j$ for $i \neq j$. We will see that our work would have been dramatically harder had we imposed these additional restrictions.

The spaces $\tilde{N}_k$ are not of any nice form (in particular, we cannot assume they are nonsingular projective, abelian varieties, and so on), so we proceed by using the p-adic point counting method described in [12], which runs in polynomial time for any variety over a field of small characteristic (that is, $p = O((d \log q)^C)$ for some positive constant C).

Theorem 5.2. *There exist an explicit deterministic algorithm and an explicit polynomial R such that for any $f \in \mathbb{F}_q[x]$ of degree d, where $q = p^m$ and p is prime, the algorithm computes the cardinality of the image set $\#(V_f)$ in a number of bit operations bounded by $R(m^d d^d p^d)$.*

Proof. We first note that

$$\begin{aligned}\tilde{N}_k &= \left\{ (x_1, \dots, x_k) \in \mathbb{F}_q^k \mid f(x_1) = \cdots = f(x_k) \right\} \\ &= \left\{ (x_1, \dots, x_k) \in \mathbb{F}_q^k \;\middle|\; \begin{matrix} f(x_1) - f(x_2) = 0 \\ f(x_1) - f(x_3) = 0 \\ \vdots \\ f(x_1) - f(x_k) = 0 \end{matrix} \right\}.\end{aligned}$$

For reasons soon to become clear, we need to represent this as the solution set of a single polynomial. Let us introduce additional variables z_1 to z_{k-1}, and set $x = (x_1, \dots, x_k)$ and $z = (z_1, \dots, z_{k-1})$. Now examine the auxiliary function

$$F_k(x, z) = z_1\big(f(x_1) - f(x_2)\big) + \cdots + z_{k-1}\big(f(x_1) - f(x_k)\big). \tag{8}$$

Clearly, if $\gamma \in \tilde{N}_k$, then $F_k(\gamma, z)$ is the zero function. If $\gamma \in \mathbb{F}_q^k \setminus \tilde{N}_k$, then the solutions of $F_k(\gamma, z) = 0$ specify a $(k-2)$-dimensional $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^{k-1}$. Thus, if we denote the cardinality of the solution set to $F_k(x, z) = 0$ as $\#(F_k)$, then we see that

$$\begin{aligned}\#(F_k) &= q^{k-1} N_k + q^{k-2}(q^k - N_k) \\ &= N_k q^{k-2}(q-1) + q^{2k-2}.\end{aligned}$$

Solving for N_k, we find that

$$N_k = \frac{\#(F_k) - q^{2k-2}}{q^{k-2}\,(q-1)}. \tag{9}$$

Thus we have an easy way to determine N_k, if we know the number of points on the hypersurface defined by the single polynomial equation $F_k = 0$.

The main theorem in [12] yields an algorithm for toric point counting in $\mathbb{F}_{q^\ell}$ that is polynomial time when the characteristic is small (that is, $p = O((d \log q)^C)$ for some positive constant C) that works for general varieties. In [12, §6.4], this theorem is adapted to be a generic point counting algorithm.

To apply this result to our problem, we note that F_k is a polynomial in $2k-1$ variables with total degree $d+1$, and that we only care about the case where $\ell = 1$. Thus, the running time for this algorithm is $\tilde{O}(2^{8k+1} m^{6k+4} k^{6k+2} d^{6k-3} p^{4k+2})$ bit operations. In order to calculate $\#(V_f)$ using (4), we calculate N_k for $1 \le k \le d$, scaled by an elementary symmetric polynomial. All of the necessary elementary symmetric polynomials can be evaluated using Newton's identities (see [15]) in $O(d^2 \log d)$ multiplications. Therefore, the entire calculation has a running time of $\tilde{O}(2^{8d+1} m^{6d+4} d^{12d-1} p^{4d+2})$ bit operations. For consistency with [12], we can then note that as $d > 1$, we can write $2^{8d+1} = d^{(\log_d 2)(8d+1)}$. Thus, there is a polynomial R in one variable such that the running time of this algorithm is bounded by $R(m^d d^d p^d)$ bit operations. In the dense polynomial model, the polynomial f has input size $O(d \log q)$, so this algorithm does not have polynomial running time with respect to the input length. This algorithm has running time that is exponential in the degree d of the polynomial, and polynomial in m and p. □

Note that if we had adopted the spaces constructed in prior works [19; 2], we would have then required $x_i \neq x_j$ for $i \neq j$. The standard approach to representing such inequalities is the "Rabinovich trick". To use this trick, we would have introduced an additional variable, say y, and the additional equation

$$y \prod_{i<j} (x_j - x_i) = 1.$$

This is a polynomial of degree $\binom{k}{2} + 1$, which would have led to an equation corresponding to (8) of degree at least $\binom{k}{2} + 2$ with $2k+1$ variables; this would have increased the work factor of the algorithm significantly.

6. Open problems

The algorithm we have presented relies on the result of Lauder and Wan, which is intended to calculate the number of $\mathbb{F}_q$-rational points on a general variety. We use this algorithm on a polynomial of a very special form. As such, it may be possible to get a considerably more efficient algorithm by exploiting symmetry in the resulting Newton polytope.

Though value sets of polynomials appear to be closely related to zero sets, they are not as well studied. There are many interesting open problems about value sets. The most important one is to find a counting algorithm with running time $(d \log q)^{O(1)}$, that is, a deterministic polynomial-time algorithm in the dense model. It is not clear if this is always possible. Our result affirmatively solves this problem for fixed d if the characteristic p is reasonably small. We conjecture that the same result is true for fixed d and all characteristic p.

For the complexity side, can one prove that the counting problem for sparse polynomials in prime-order finite fields is hard? Can one prove that the counting problem for the dense input model is hard for general degree d?

Acknowledgments

We thank Dr. Tsuyoshi Ito for pointing out reference [6] to us. All three authors are partially supported by the NSF.

References

[1] Martín Avendaño, Ashraf Ibrahim, J. Maurice Rojas, and Korben Rusek, *Randomized NP-completeness for p-adic rational roots of sparse polynomials in one variable*, in Watt [21], 2010, pp. 331–338. MR 2920572

[2] B. J. Birch and H. P. F. Swinnerton-Dyer, *Note on a problem of Chowla*, Acta Arith. **5** (1959), 417–423. MR 22 #4675

[3] J. P. Buhler and P. Stevenhagen (eds.), *Algorithmic number theory: lattices, number fields, curves and cryptography*, Mathematical Sciences Research Institute Publications, no. 44, Cambridge University Press, 2008. MR 2009h:11003

[4] Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung (eds.), *Automata, languages and programming: Proceedings of the 32nd International Colloquium (ICALP 2005) held in Lisbon, July 11–15, 2005*, Lecture Notes in Computer Science, no. 3580, Berlin, Springer, 2005. MR 2006f:68001

[5] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to algorithms*, 2nd ed., MIT Press, Cambridge, MA, 2001. MR 2002e:68001

[6] B. Durand, *Inversion of 2D cellular automata: some complexity results*, Theoret. Comput. Sci. **134** (1994), no. 2, 387–401. MR 96b:68131

[7] Erich Kaltofen, *Polynomial factorization: a success story*, slides presented at the International Symposium on Symbolic and Algebraic Computation (ISSAC ’03), Philadelphia, August 3–6, 2003. http://www4.ncsu.edu/~kaltofen/bibliography/lectures/lectures.html#issacphiladelphia

[8] Erich Kaltofen and Pascal Koiran, *On the complexity of factoring bivariate supersparse (lacunary) polynomials*, in Kauers [9], 2005, pp. 208–215. MR 2280549

[9] Manuel Kauers (ed.), *ISSAC’05: Proceedings of the 30th International Symposium on Symbolic and Algebraic Computation held in Beijing, July 24–27, 2005*, New York, ACM Press, 2005, held in Beijing, July 24–27, 2005. MR 2007g:68005

[10] Neeraj Kayal, *Solvability of a system of bivariate polynomial equations over a finite field (extended abstract)*, in Caires et al. [4], 2005, pp. 551–562. MR 2184660

[11] Aviad Kipnis and Adi Shamir, *Cryptanalysis of the HFE public key cryptosystem by relinearization*, in Wiener [22], 1999, pp. 19–30. MR 2000i:94052

[12] Alan G. B. Lauder and Daqing Wan, *Counting points on varieties over finite fields of small characteristic*, in Buhler and Stevenhagen [3], 2008, pp. 579–612. MR 2009j:14029

[13] Keju Ma and Joachim von zur Gathen, *The computational complexity of recognizing permutation functions*, Comput. Complexity **5** (1995), no. 1, 76–97. MR 96c:68066

[14] ———, *Tests for permutation functions*, Finite Fields Appl. **1** (1995), 31–56. MR 96a:11137

[15] D. G. Mead, *Newton's identities*, Amer. Math. Monthly **99** (1992), no. 8, 749–751. MR 93h:05011

[16] René Peralta, *On the distribution of quadratic residues and nonresidues modulo a prime number*, Math. Comp. **58** (1992), no. 197, 433–440. MR 93c:11115

[17] I. E. Shparlinski, *A deterministic test for permutation polynomials*, Comput. Complexity **2** (1992), no. 2, 129–132. MR 93h:11136

[18] Terence Tao, Ernest Croot, III, and Harald Helfgott, *Deterministic methods to find primes*, Math. Comp. **81** (2012), no. 278, 1233–1246. MR 2869058

[19] Saburô Uchiyama, *Note on the mean value of $V(f)$*, Proc. Japan Acad. **31** (1955), 199–201. MR 17,130f

[20] Joachim von zur Gathen, *Tests for permutation polynomials*, SIAM J. Comput. **20** (1991), no. 3, 591–602. MR 92g:11117

[21] Stephen M. Watt (ed.), *ISSAC 2010—Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, Association for Computing Machinery, New York, 2010. MR 2920530

[22] Michael Wiener (ed.), *Advances in cryptology—CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference held in Santa Barbara, CA, August 15–19, 1999*, Lecture Notes in Computer Science, no. 1666, Berlin, Springer, 1999. MR 2000h:94003

QI CHENG: qcheng@cs.ou.edu
School of Computer Science, The University of Oklahoma, Norman, OK 73019, United States

JOSHUA E. HILL: hillje@math.uci.edu
Department of Mathematics, University of California, Irvine, Irvine, CA 92697, United States

DAQING WAN: dwan@math.uci.edu
Department of Mathematics, University of California, Irvine, Irvine, CA 92697, United States

VOLUME EDITORS

Everett W. Howe
Center for Communications Research
4320 Westerra Court
San Diego, CA 92121-1969
United States

Kiran S. Kedlaya
Department of Mathematics
University of California, San Diego
9500 Gilman Drive #0112
La Jolla, CA 92093-0112

Front cover artwork based on a detail of
Chicano Legacy 40 Años ©2010 Mario Torero.

Electronic copies can be obtained free of charge from http://msp.org/obs/1 and printed copies can be ordered from MSP (contact@msp.org).

The Open Book Series is a trademark of Mathematical Sciences Publishers.

ISSN: 2329-9061 (print), 2329-907X (electronic)

ISBN: 978-1-935107-00-2 (print), 978-1-935107-01-9 (electronic)

First published 2013.

msp

MATHEMATICAL SCIENCES PUBLISHERS
798 Evans Hall #3840, c/o University of California, Berkeley CA 94720-3840
contact@msp.org http://msp.org

THE OPEN BOOK SERIES 1

Tenth Algorithmic Number Theory Symposium

The Algorithmic Number Theory Symposium (ANTS), held biennially since 1994, is the premier international forum for research in computational number theory. ANTS is devoted to algorithmic aspects of number theory, including elementary, algebraic, and analytic number theory, the geometry of numbers, arithmetic algebraic geometry, the theory of finite fields, and cryptography.

This volume is the proceedings of the tenth ANTS meeting, held July 9–13, 2012, at the University of California, San Diego. It includes revised and edited versions of the 25 refereed papers presented at the conference, together with extended abstracts of two of the five invited talks.

TABLE OF CONTENTS