Fast Computation of Roots of Polynomials over Function Fields and Fast List Decoding of Algebraic Geometric Codes [*]

Xin-Wen Wu, Paul H. Siegel

Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093-0407
Email: wxw@cwc.ucsd.edu, and psiegel@cwc.ucsd.edu

(Extended Summary)

1 Introduction

Suppose C is a $[n, k, d]$ code over $\mathbf{F}_q$, $t < n$ is a positive integer. For any received vector $\mathbf{y} = (y_1, \cdots, y_n) \in \mathbf{F}_q^n$, we refer to any code word $\mathbf{c}$ in C satisfying $d(\mathbf{c}, \mathbf{y}) \leq t$ as a *t-consistent code word*. A decoding problem is in fact the problem of finding an effective (or efficient) algorithm which can find t-consistent code words, and we call such an algorithm a decoding algorithm that can correct t errors.

The *classical decodings* (or call *unique decodings*) only consider the algorithms which can correct $\tau = \lfloor \frac{d-1}{2} \rfloor$ or fewer errors. It is clear that in any Hamming sphere in $\mathbf{F}_q^n$ of radius $\leq \tau$, there exist at most one code word of a $[n, k, d]$ code. We call τ the *error correction bound* of the code. On the other hand, if the number of errors $t \geq \tau$ then there may exist several different consistent code words. A *list decoding* is a decoding algorithm which tries to construct a list of all consistent code words. Thus, a list decoding algorithm makes it is possible to recover the information from errors beyond the traditional error correction bound.

The list decoding problem was first defined by Elias [2]. In [11], Sudan proposed a list decoding algorithm for generalized Reed-Solomon codes. Shokrollahi and Wasserman generalized Sudan's algorithm and derived a list decoding scheme for algebraic geometric codes [10]. However, for codes of higher rates, these algorithms do not improve the classical decoding algorithms, i.e., these algorithms are effective only for low rate codes.

In a very recent paper [4], Guruswami and Sudan proposed an improved polynomial-time algorithm for Reed-Solomon and algebraic geometric codes. The algorithm has a better error-correction rate than well-known algorithms for every choice of the code rates.

[*] This work was supported in part by Grant No. NCR-9612802 from the National Science Foundation (NSF) and by a research grant from the National Storage Industry Consortium (NSIC).

However, Guruswami and Sudan's algorithm is rather complicated, especially for algebraic geometric codes. In their paper [8], Roth and Ruckenstein improved Sudan's previous work in [11], and presented a fast list decoding algorithm for Reed-Solomon codes.

In this paper, motivated by Roth and Ruckenstein's work, we derive a fast list decoding algorithm for algebraic geometric codes. Both Shokrollahi and Wasserman's algorithm and Guruswami and Sudan's algorithm are based on the factorization algorithm (or root-finding algorithm) of polynomials in $\mathcal{K}[T]$, where $\mathcal{K}$ is the function field of some algebraic curve, and T is a formal variable. Shokrollahi and Wasserman [10] proposed a polynomial time algorithm for the factorization, but it is rather arduous. Our improved decoding scheme is based on an efficient algorithm of finding roots of polynomials over function fields that we will propose in Section 3, and does not need the factorization algorithm. The new algorithm achieves a lower complexity.

2 Algebraic Geometric Codes and List Decodings

Let $\mathcal{X} \subseteq \mathbf{PF}_q^m$ be a nonsingular, irreducible curve in m-dimensional projective space, of genus g, defined by the following projective equation

$$F(X) = F(X_1, X_1, \cdots, X_{m+1}) = 0.$$

Suppose $\{P_1, P_2, \cdots, P_n\}$ is a set of rational points of $\mathcal{X}$. Let $D = P_1 + \cdots + P_n$, and G be another divisor on $\mathcal{X}$ satisfying $\sup(D) \cap \sup(G) = \emptyset$. An algebraic geometric code (AG code) $C_L(D, G)$ is defined as

$$C_L(D, G) = \{(f(P_1), f(P_2), \cdots, f(P_n)) \mid f \in L(G)\},$$

where $L(G) = \{f \text{ is in the function field } \mathcal{K} \text{ of } \mathcal{X} \mid f = 0 \text{ or } (f) + G \geq 0\}$. Suppose $\rho = \deg G < n$, then $C_L(D, G)$ has length n, dimension $\geq \rho - g + 1$, and minimum distance $\geq n - \rho$.

Consider the algebraic geometric codes $C_L(D, G)$ with $D = P_1 + \cdots + P_n$ and $G = \rho P$, where ρ is an integer and $\{P_1, \cdots, P_n, P\}$ is the set of all rational points of $\mathcal{X}$. It is clear that these algebraic geometric codes achieve the longest code length, thus, they are of special interest in practical applications. Let ρ be a nonnegative integer. If there exists a rational function $\varphi \in \mathcal{K}$, such that φ has pole only at P and the order of pole of φ at P is ρ, i.e., $\mathrm{ord}_P(\varphi) = -\rho$, then we call ρ a *nongap* of P. Let $\{\rho_1, \rho_2, \rho_3, \cdots\}$ be the set of all nongaps of P and $\rho_1 < \rho_2 < \rho_3 < \cdots$. Then, it is well known that

$$0 = \rho_1 < \rho_2 < \cdots < \rho_g < \rho_{g+1} = 2g,$$

and $\rho_i = i + g - 1$ when $i \geq g + 1$.

Let $\varphi_1, \varphi_2, \varphi_3, \cdots$ be a sequence of rational functions, such that φ_i has pole only at P and $\mathrm{ord}_P(\varphi_i) = -\rho_i$. Then it is easy to check that $\{\varphi_1, \varphi_2, \cdots, \varphi_i\}$ is a basis of $L(\rho_i P)$.

For the purpose of completeness, we restate Guruswami and Sudan's decoding algorithm by the classical notation of algebraic geometric codes as following.

List Decoding Algorithm for AG Codes

Implicit Parameters: $n; P, P_1, \cdots, P_n \in \mathcal{X}$; $k; g$.

Assumptions: Assume that we know a basis $\{\varphi_{j_1} \mid j_1 = 1, \cdots, l-g+1\}$ of $L(lP)$; and for every P_i, a sequence of rational functions $\{\psi_{j_3,P_i} \mid j_3 = 1, \cdots, l-g+1\}$, such that $\mathrm{ord}_{P_i}(\psi_{j_3,P_i}) \geq j_3 - 1$. And assume we know a set of coefficients $\{\alpha_{P_i,j_1,j_3} \in \mathbf{F}_q \mid i = 1, \cdots, n;\ j_1, j_3 = 1, \cdots, l-g+1\}$ such that for every i, j_1

$$\varphi_{j_1} = \sum_{j_3=1}^{l-g+1} \alpha_{P_i,j_1,j_3} \psi_{j_3,P_i}.$$

Step 0: Compute parameters r, l such that

$$rt > l \quad \text{and} \quad \frac{(l-g)(l-g+1)}{2\rho} > n \left(\begin{array}{c} r+1 \\ 2 \end{array} \right),$$

where $\rho := k + g - 1$.

Step 1: Find a nonzero polynomial $H(T) \in \mathcal{K}[T]$ of the form

$$H(T) = H(X;T) = \sum_{j_2=0}^{s} h_{j_2}(X) T^{j_2},$$

where $s := \lfloor \frac{l-g}{\rho} \rfloor$, $h_{j_2}(X) \in L((l - j_2\rho)P)$, and $h_{j_2}(X) = \sum_{j_1=1}^{l-j_2\rho-g+1} h_{j_1 j_2} \varphi_{j_1}(X)$, such that for $i = 1, \cdots, n$; $j_3 \geq 1$, $j_4 \geq 0$, and $j_3 + j_4 \leq r$,

$$h^{(i)}_{j_3,j_4} := \sum_{j_2=j_4}^{s} \sum_{j_1=1}^{l-j_2\rho-g+1} \left(\begin{array}{c} j_2 \\ j_4 \end{array} \right) y_i^{j_2-j_4} h_{j_1,j_2} \alpha_{P_i,j_1,j_3} = 0.$$

Step 2: Using the algorithm of factorization (or root-finding), find all roots $f \in L((k + g - 1)P) = L(\rho P)$ of the polynomial $H(T)$. For each such f, check if $f(P_i) = y_i$ for at least t values of $i \in \{1, \cdots, n\}$, and if so, include f in output list.

For the complexity of the above algorithm, by Proposition 22 in [4], all the α_{P_i,j_1,j_3}'s can be found in $O(nl^2)$ operations over $\mathcal{K}$. Also, the nonzero polynomial $H(T)$ can be found by Gaussian eliminations in $O(l^6/\rho^3)$ operations (over $\mathbf{F}_q$). So the complexity is mainly based on the factorization algorithm (Step 2). According to Shokrollahi and Wasserman [10], there is a factorization algorithm that runs in time polynomial in the representation of the field. However, it is not still clear if these representations are of size that is bounded by some polynomial in the length of the codes.

3 Fast Algorithm of Finding Roots of $H(T)$

In this section, we will derive an efficient algorithm to find the T-roots of $H(X;T) = 0$ in $L(G)$, which does not need factorization of $H(X;T)$.

In our case, $G = \rho P$, and $H(X;T) = h_0(X) + h_1(X)T + \cdots + h_s(X)T^s$, where

$$h_j(X) \in L((l - j\rho)P), \qquad j = 0, 1, \cdots, s,$$

where $L(lP) \supseteq L((l-\rho)P) \supseteq L((l-2\rho)P) \supseteq \cdots \supseteq L((l-s\rho)P)$. Suppose $f(X) \in L(G) = L(\rho P)$ such that $H(X; f(X)) = 0$. Let $\{\varphi_1, \varphi_2, \cdots, \varphi_k\}$ be a basis of $L(G)$, we can assume

$$f(X) = f_1\varphi_1(X) + f_2\varphi_2(X) + \cdots + f_k\varphi_k(X),$$

where $f_1, f_2, \cdots, f_k \in \mathbf{F}_q$. We now can find f_k, f_{k-1}, $\cdots$, f_1, by the following k steps.

STEP 1 (to find f_k). Set $G_1(X;T) = H_1(X;T) = H(X;T)$ and $\widehat{G}_1(X;T) = G_1(X;\varphi_k T)$. Then,

$$\widehat{G}_1(X;T) = h_0 + (h_1\varphi_k)T + \cdots + (h_j\varphi_k^j)T^j + \cdots + (h_s\varphi_k^s)T^s.$$

Since $h_j \in L((l - j\rho)P)$ and $\varphi_k \in L(\rho P)$, we have

$$h_j\varphi_k^j \in L(lP), \qquad j = 0, 1, \cdots, s,$$

i.e., $\mathrm{ord}_P(h_j\varphi_k^j) \geq -l$. Let

$$-\rho_{r_1} = \min\{\mathrm{ord}_P(h_j\varphi_k^j) \mid j = 0, 1, \cdots, s\}.$$

Suppose φ_{r_1} is a rational function and $\mathrm{ord}_P(\varphi_{r_1}) = -\rho_{r_1}$. Divide $\widehat{G}_1(X;T)$ by φ_{r_1}, and let

$$\widetilde{G}_1(X;T) = \frac{1}{\varphi_{r_1}}\widehat{G}_1(X;T) = \frac{h_0}{\varphi_{r_1}} + \left(\frac{h_1\varphi_k}{\varphi_{r_1}}\right)T + \cdots + \left(\frac{h_s\varphi_k^s}{\varphi_{r_1}}\right)T^s.$$

Then, by the definition of φ_{r_1},

$$\left(\frac{h_0}{\varphi_{r_1}}(P), \left(\frac{h_1\varphi_k}{\varphi_{r_1}}\right)(P), \cdots, \left(\frac{h_s\varphi_k^s}{\varphi_{r_1}}\right)(P)\right) \in \mathbf{F}_q^{s+1} - \{\mathbf{0}\}. \tag{3.1}$$

This means that $\widetilde{G}_1(P;T)$ is a nonzero polynimial in $\mathbf{F}_q[T]$.

On the other hand, by $H(X; f(X))$ =0, we have

$$\widehat{G}_1(X; \frac{f(X)}{\varphi_k(X)}) = 0, \tag{3.2}$$

and

$$\widetilde{G}_1(X; \frac{f(X)}{\varphi_k(X)}) = 0. \tag{3.3}$$

Since $\mathrm{ord}_P\left(\frac{\varphi_j}{\varphi_k}\right) = \rho_k - \rho_j > 0$, for $j = 1, \cdots, k-1$, we have $\frac{\varphi_j}{\varphi_k}(P) = 0$. Thus,

$$\frac{f}{\varphi_k}(P) = f_1\frac{\varphi_1}{\varphi_k}(P) + \cdots + f_{k-1}\frac{\varphi_{k-1}}{\varphi_k}(P) + f_k = f_k.$$

By (3.3), we have

$$\widetilde{G}_1(P; f_k) = \widetilde{G}_1(P; \frac{f}{\varphi_k}(P)) = \widetilde{G}_1(X; \frac{f(X)}{\varphi_k(X)})(P) = 0. \tag{3.4}$$

It follows that f_k is a root of the nonzero polynomial equation $\widetilde{G}_1(P;T)=0$ over $\mathbf{F}_q$.

Suppose we have executed STEP 1, $\cdots$, STEP i, and in STEP i, we have a polynomial

$$\widetilde{G}_i(X;T)=\widetilde{G}_i^{(0)}(X)+\widetilde{G}_i^{(1)}(X)T+\cdots+\widetilde{G}_i^{(s)}(X)T^s,$$

such that

$$\widetilde{G}_i(X;\frac{f^{(i)}(X)}{\varphi_{k-i+1}(X)})=0, \tag{3.5}$$

and $\widetilde{G}_i(P;T)$ is a nonzero polynomial in $\mathbf{F}_q[T]$, and f_{k-i+1} is a solution of $\widetilde{G}_i(P;T)=0$. We now enter STEP i+1.

STEP i+1 (to find f_{k-i}). Set $G_{i+1}(X;T)=\widetilde{G}_i(X;T+f_{k-i+1})$ and $\widehat{G}_{i+1}(X;T)=G_{i+1}(X;\frac{\varphi_{k-i}}{\varphi_{k-i+1}}T)$. Suppose

$$G_{i+1}(X;T)=G_{i+1}^{(0)}(X)+G_{i+1}^{(1)}(X)T+\cdots+G_{i+1}^{(s)}(X)T^s,$$

and

$$\widehat{G}_{i+1}(X;T)=\widehat{G}_{i+1}^{(0)}(X)+\widehat{G}_{i+1}^{(1)}(X)T+\cdots+\widehat{G}_{i+1}^{(s)}(X)T^s.$$

It is easy to prove that $G_{i+1}^{(j)}(P)$, $\widehat{G}_{i+1}^{(j)}(P)\in\mathbf{F}_q$, for $j=0,1,\cdots,s$, and $\mathrm{ord}_P(\widehat{G}_{i+1}^{(j)}(X))>0$ for $j=1,2,\cdots,s$. Let

$$\rho_{r_{i+1}}=\min\{\mathrm{ord}_P(\widehat{G}_{i+1}^{(j)}(X))\mid j=0,1,\cdots,s\}.$$

Suppose $\varphi_{r_{i+1}}$ be a rational function such that $\mathrm{ord}_P(\varphi_{r_{i+1}})=\rho_{r_{i+1}}$. Divide $\widehat{G}_{i+1}(X;T)$ by $\varphi_{r_{i+1}}$,

$$\widetilde{G}_{i+1}(X;T)=\frac{1}{\varphi_{r_{i+1}}}\widehat{G}_{i+1}(X;T)$$

Then, $\widetilde{G}_{i+1}(P;T)$ is a nonzero polynomial in $\mathbf{F}_q[T]$.

Let $f^{(i+1)}(X)=f^{(i)}(X)-f_{k-i+1}\varphi_{k-i+1}(X)=f_1\varphi_1(X)+\cdots+f_{k-i}\varphi_{k-i}(X)$, we have

$$\widehat{G}_{i+1}(X;\frac{f^{(i+1)}(X)}{\varphi_{k-i}(X)})=G_{i+1}(X;\frac{f^{(i+1)}(X)}{\varphi_{k-i+1}(X)})=\widetilde{G}_i(X;\frac{f^{(i)}(X)}{\varphi_{k-i+1}(X)})=0.$$

Thus,

$$\widetilde{G}_{i+1}(X;\frac{f^{(i+1)}(X)}{\varphi_{k-i}(X)})=0. \tag{3.6}$$

Since $\frac{f^{(i+1)}(P)}{\varphi_{k-i}(P)}=f_{k-i}$, we have

$$\widetilde{G}_{i+1}(P;f_{k-i})=0.$$

So, f_{k-i} is a solution of the nonzero polynomial equation $\widetilde{G}_{i+1}(P;T)=0$.

In the above discussion, we in fact obtain an iterative algorithm to find the roots $f(X)$ of $H(T)$ by finding the coefficients of $f(X)$ step by step.

Lemma 3.1 *Suppose f_{k-i+1} is a solution of multiplicity h of the nonzero equation $\widetilde{G}_i(P;T) = 0$ in STEP i, and $\widetilde{G}_{i+1}(P;T) = 0$ is the nonzero equation in STEP i+1. Then, $\deg_T \widetilde{G}_{i+1}(P;T) \leq h$.*

Theorem 3.1 *Let $f(X) = f_1\varphi_1(X) + \cdots + f_k\varphi_k(X) \in L(G)$ and $H(X; f(X)) = 0$. Then we can get the coefficients of $f(X)$ by solving polynomial equations of degree s and with coefficients in $\mathbf{F}_q$ step by step. Moreover, we have at most sk possible solutions for $(f_1, f_2, \cdots, f_k)$.*

Lemma 3.2 ([6]) *The roots in $\mathbf{F}_q$ of a polynomial of degree s can be found in time complexity $O((s^2 \log^2 s) \log q)$.*

Theorem 3.2 *Given a nonzero polynomial $H(T)$ over $\mathcal{K}[T]$ of degree s returned in Step 1 of the List Decoding Algorithm. Then the roots of $H(T)$ in $L(G) = L((k+g-1)P)$ can be found in $O(k(s^2 \log^2 s) \log q)$ operations over $\mathbf{F}_q$ and $O(ks^2)$ operations over $\mathcal{K}$.*

Replacing the factorization algorithm (Step 2 in Guruswami and Sudan's algorithm) by the above algorithm, we obtain a fast list decoding algorithm of algebraic geometric codes.

References

[1] I. Blake, C. Heegard, T. Hϕholdt, and V. Wei, "Algebraic -geometry codes", *IEEE Trans. on Inform. Theory*, Vol.44, No.6, 1998, pp. 2596-2618.

[2] P. Elias, "List decoding for noisy channel", *Tech. Rep. 335, Res. Lab. Electron.*, MIT, Cambridge, MA, 1957.

[3] P. Elias, "Error-correcting codes for list decoding", *IEEE Trans. on Inform. Theory*, Vol.37, No.6, 1991, pp. 5-12.

[4] V. Guruswami, M. Sudan, "Improved Decoding of Reed-Solomon and Algebraic-Geometry Codes", *IEEE Trans. on Inform. Theory*, Vol.45, No.6, 1999, pp. 1757-1767.

[5] T. Hϕholdt, R. Pellikaan, "On the decoding of algebraic-geometric codes", *IEEE Trans. on Inform. Theory*, Vol.41, No.6, 1995, pp. 1589-1614.

[6] R. Lidl, H. Niederreiter, *Finite Fields*, Addison-Wesley, Reading, Massachusetts, 1983.

[7] M.O. Rabin, "Probabilistic algorithms in finite fields", *SIAM J. Compute.*, 9, 1980, pp.273-280.

[8] R. Roth, G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance", preprint, 1998.

[9] M. Shokrollahi, H. Wasserman, "Decoding algebraic-geometric codes", in *Proc. IEEE Workshop Inform. Theory 1998*, Killarney, Ireland.

[10] M. Shokrollahi, H. Wasserman, "List decoding of algebraic-geometric codes", *IEEE Trans. on Inform. Theory*, Vol.45, No.2, 1999, pp. 432-437.

[11] M. Sudan, "Decoding of Reed Solomon codes beyond the error-correction bound", *Journal of Complexity*, 13, 1997, pp. 180-193.