On the Interpolation Step in the Guruswami-Sudan List Decoding Algorithm for Reed-Solomon Codes

P. V. Trifonov
Saint-Petersburg State Polytechnic University, Distributed Computing and Networking Department
e-mail: petert@dcn.nord.nw.ru

Abstract

Divide-and-conquer method for interpolation in Guruswami-Sudan (GS) list decoding algorithm is considered. It is shown that the Groebner basis (GB) of the ideal of bivariate interpolation polynomials (IP) can be obtained as a product of trivariate polynomials corresponding to disjoint subsets of interpolation points. The impact of monomial ordering on the interpolation complexity is studied.

1 Introduction

A major drawback of the GS list decoding algorithm [1] is its high complexity. This paper addresses the problem of efficient implementation of the bivariate interpolation step in the GS algorithm, which is known to be the most complex part of it. This step consists in construction of a bivariate polynomial with pre-assigned zeroes of a given multiplicity, which is equivalent to solving a homogeneous system of linear equations. A number of reduced-complexity interpolation algorithms exploiting the structure of this system were developed [2, 3]. Essentially, these algorithms iteratively construct the set of candidate IPs. The polynomials continuously grow during the computation, making the complexity to be quadratic in the number of equations. In some cases the number of equations may be reduced considerably [4].

The standard way to reduce the complexity of such algorithms is the divide-and-conquer approach. The main idea of this method was introduced in [5]. However, some implementation issues were not solved at that time. This paper addresses them by introducing an efficient method for construction of the GB of the ideal of IPs.

The paper is organized as follows. Section 2 presents an overview of the GS and related algorithms. Section 3 describes the proposed interpolation method. Finally, conclusions are drawn in Section 4.

2 List decoding of Reed-Solomon codes

2.1 Guruswami-Sudan algorithm

The GS algorithm solves the problem of list decoding of $(n, k+1, n-k)$ RS code defined over field $\mathbb{F}$, i.e. finding for any given vector $(y_1, \ldots, y_n) \in \mathbb{F}^n$ all codewords matching with it in at least τ positions. The problem can be reformulated as finding all polynomials $f_{(j)}(x) : \deg f_{(j)}(x) \leq k, |\{x_i | f_{(j)}(x_i) = y_i\}| \geq \tau$, where $x_i \in \mathbb{F}$ are code locators.

Definition 1. j-th Hasse derivative (HD) $g^{[j]}(x_0)$ of a polynomial $g(x) = \sum_{i=0}^{t} g_i x^i$ at point x_0 is equal to the j-th coefficient of a "shifted" polynomial $g(x+x_0) = \sum_{i=0}^{t} g'_i x^i$. Alternatively, $g^{[j]}(x_0) = \frac{1}{j!} g^{(j)}(x_0)$, where $g^{(j)}(x)$ is the conventional formal derivative of $g(x)$.

Similarly one can define partial HDs of a bivariate polynomial. A polynomial is said to have a root of multiplicity r at point z if all its HDs at z of total order less than r are equal to zero.

The GS algorithm constructs a bivariate polynomial $Q(x,y)$ with a minimal possible $(1,k)$-weighted degree (see [1]) having all points (x_i,y_i) as roots of multiplicity r, and then finds all $f_{(j)}(x)$ as its roots:

1. Construct a polynomial $Q(x,y) : \mathrm{wdeg}_{(1,\mathrm{k})} Q(x,y) \le l$, where $\mathrm{wdeg}_{(a,b)} Q(x,y)$ denotes (a,b)-weighted degree of polynomial $Q(x,y)$, such that
$$Q^{[j_1,j_2]}(x_i,y_i) = 0, j_1+j_2 < r, i = 1..n \quad (1)$$
2. Find all polynomials $f_{(j)}(x) : \deg f_{(j)}(x) \le k$, s.t. $Q(x, f_{(j)}(x)) = 0$. Select among them the solutions of the list decoding problem.

Correctness proof of the algorithm and instructions for selecting its parameters l and r can be found in [1, 2]. Efficient algorithm for finding roots of a bivariate polynomial is presented in [6].

2.2 Implementing the interpolation step

The iterative interpolation algorithm presented in [2] exploits the structure of the system of linear equations (1). Essentially, it constructs a GB of the module of IPs satisfying $\mathrm{wdeg}_{(0,1)} Q(x,y) \le \rho$ for some ρ [5]. That is, it constructs a $((\rho+1)\times(\rho+1)$ matrix polynomial $Q(x)$ (interpolating matrix) so that any IP satisfying the above constraints can be represented as $Q(x,y) = YQ(x)(p_0(x),\ldots,p_\rho(x))^T$ for some polynomials $p_i(x)$, where $Y = (y^0,\ldots,y^\rho)$. The algorithm starts from $Q(x) = I$ and processes all interpolation equations (1). For each equation it constructs a matrix

$$\Delta^{(i,j_1,j_2)} = \begin{pmatrix} 1 & 0 & \ldots & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & \ldots & 0 \\ \ldots & & & & & \\ -\frac{\Delta_0}{\Delta_{j_0}} & -\frac{\Delta_1}{\Delta_{j_0}} & \ldots & x-x_i & \ldots & -\frac{\Delta_r}{\Delta_{j_0}} \\ \ldots & & & & & \\ 0 & 0 & \ldots & 0 & \ldots & 1 \end{pmatrix}$$

and multiplies $Q(x)$ by it. Observe that $\det \Delta^{(i,j_1,j_2)} = \delta(x-x_i), \delta \neq 0$. Hence, for any algebraic extension of $\mathbb{F}$ the matrix polynomial $Q(x)$ obtained after processing all interpolation constraints is singular only for $x = x_i, i = 1..n$. Furthermore, its kernel at these points is also finite-dimensional. This implies that the ideal generated by components of $(Q_0(x,y),\ldots,Q_\rho(x,y)) = YQ(x)$ is a zero-dimensional one [7].

This property allows one to perform interpolation as follows [5]. Partition the set of interpolation points $V \subset \mathbb{F}^2$ into two disjoint subsets $V_0, V_1 : V_0 \bigcup V_1 = V$. Construct the interpolating matrices $Q^{(0)}(x)$ and $Q^{(1)}(x)$ for each subset and associated ideals I_0 and I_1, $I_s =< Q_0^{(s)}(x,y),\ldots,Q_\rho^{(s)}(x,y) >, s = 0,1$. By construction, these ideals are coprime, and the ideal of IPs satisfying (1) is given by a product of I_0 and I_1, i.e. $I = I_0 I_1 =< Q_i^{(0)}(x,y)Q_j^{(1)}(x,y), 0 \le i,j \le \rho >$.

For any monomial ordering the minimal IP always appears in the reduced GB of the ideal of IPs [8]. It is therefore necessary to apply the Buchberger algorithm in order to find all the solutions of the list decoding problem. The input to this algorithm is the set of all pairwise products $Q_i^{(0)}(x,y)Q_j^{(1)}(x,y)$. Usually the reduced GB has much less elements, and they have much smaller degree. This means that straightforward ideal multiplication with subsequent Buchberger reduction is not an optimal method.

3 Constructing the product of ideals

3.1 Fast ideal multiplication

This section introduces a fast method for computing the product of ideals of IPs, which is based on the one-to-one correspondence between zero-dimensional ideals and zeroes of Hasse derivatives of their bases.

Definition 2 ([9])**.** Let $D^{[i,j]}$ be the differential operator corresponding to computing HD of order i over variable x and order j over y. Let

$$\sigma_{x^l y^m}(D^{[i,j]}) = \begin{cases} D^{[i-l,j-m]}, & i \ge l \wedge j \ge m \\ 0, & \text{otherwise.} \end{cases}$$

Subset G of the set of differential operators $\mathbb{D}$ is closed, if $\forall (l,m) \in \mathbb{N}^2, \forall \delta \in G : \sigma_{x^l y^m}(\delta) \in G$.

Theorem 1 ([9], Th. 2.8)**.** *Every zero-dimensional ideal I is uniquely defined by giving a set of points $t_1, \ldots, t_n$ in $\mathbb{F}$, and for each point a closed subspace $G_i = \mathrm{span}_{\mathbb{F}}(\delta_{i1}, \ldots, \delta_{i,s_i}) \subset \mathrm{span}_{\mathbb{F}}(\mathbb{D})$, so that $f \in I$ iff $\forall i,j : \delta_{ij}(t_i)(f) = 0$, where $\delta_{ij}(t_i)(f)$ is the value of differential operator δ_{ij} applied to f at point t_i.*

The set G of HDs of total order less than r is closed, so the ideal of IPs is unique.

Definition 3. Let $\{Q_0(x,y), \ldots, Q_\rho(x,y)\} \subset \mathbb{F}[x,y]$ be a basis of ideal I. The generating function of the basis B is defined as $Q(x,y,z) = \sum_{i=0}^{\rho} Q_i(x,y) z^i$.

The root set of a generating function of a basis of ideal I includes $V(I) \times \mathbb{F}$, where $V(I)$ is the affine variety of I, as well as some other points, which depend on the basis used and order of elements in it.

Theorem 2. *Let $I_s =< Q_0^{(s)}(x,y), \ldots, Q_\rho^{(s)}(x,y) >, s = 0,1$ be zero-dimensional coprime ideals. Then $I' =< \sum_{j=0}^{\rho} Q_{i-j}^{(0)}(x,y) Q_j^{(1)}(x,y), i = 0..2\rho >$ and $I = I_0 I_1$ coincide.*

Proof. Let $V_s = V(I_s) = \{(x_i, y_i)\}$ be the set of zeroes of I_s, and let G_i be the associated closed sets of nullified differential operators. The generating function of any I_s basis may be represented as

$$\forall (x_i,y_i) \in V_s : Q^{(s)}(x,y,z) = \sum_l Q_l^{(s)}(x,y) z^l = \sum_{(j_1,j_2): D^{[j_1,j_2]} \notin G_i} (x-x_i)^{j_1} (y-y_i)^{j_2} P_{ij_1j_2}(x,y,z),$$

where $P_{ij_1j_2}(x,y,z)$ are some polynomials such that $P_{ij_1j_2}(x_i,y_i,z) \neq 0$. Since I_0 and I_1 are coprime, $(x_i,y_i) \in V_0$ implies $Q^{(1)}(x_i,y_i,z) \neq 0$ and vice versa. Hense, the product of generating functions $Q(x,y,z) = Q^{(0)}(x,y,z) Q^{(1)}(x,y,z)$ satisfies

$$Q(x,y,z) \quad = \sum_{(j_1,j_2): D^{[j_1,j_2]} \notin G_i} (x-x_i)^{j_1} (y-y_i)^{j_2} P_{ij_1j_2}(x,y,z) Q^{(1-s)}(x,y,z), (x_i,y_i) \in V_s, s = 0,1$$

and $P_{ij_1j_2}(x_i,y_i,z) Q^{(1-s)}(x_i,y_i,z) \neq 0, (x_i,y_i) \in V_s$, i.e. algebraic roots multiplicity of all polynomials in I' is exactly the same as in original ideals. Hence,

$$Q \in I' \Leftrightarrow (\forall (x_i,y_i) \in V_0 \bigcup V_1) \delta_{ij}(x_i,y_i)(Q) = 0, \tag{2}$$

where differential operators δ_{ij} are exactly the same as for I_0 and I_1. Since all pairs $Q_i^{(0)}(x,y) Q_j^{(1)}(x,y)$ belong both to I_0 and I_1, they satisfy the constraint (2). Hence, they belong to I', that is $I \subset I'$. Inclusion $I' \subset I$ is obvious. □

This theorem enables one to compute the product of ideals by multiplying the generating functions of their bases. This can be implemented by any fast linear convolution algorithm. Furthermore, the basis of ideal product obtained in this way contains only $2\rho + 1$ elements, which is much less than $(\rho+1)^2$ for the case of conventional ideal multiplication rule.

3.2 Extracting module from the ideal

Recall, that list decoding of RS codes requires one to construct an IP with minimal possible weighted degree. It is known to appear in the GB of the module of IPs with y-degree not higher than ρ [2]. Since the ideal of bivariate IPs can be considered as an infinite-dimensional module, so that any polynomial in it is represented as $(y^0, y^1, \ldots) Q(x) (p_0(x), p_1(x), \ldots)^T$, one has to project it onto $\rho + 1$ first coordinates. This can be performed by triangularizing the semi-infinite matrix $Q(x)$. However, this can be avoided by using the Groebner bases of I_0 and I_1 constructed for lexicographic monomial ordering. In this case leading terms of basis elements can be written as $\mathrm{LT}\, Q_i^{(s)}(x,y) = a x^b y^i$ for some $a \in \mathbb{F}, b \in \mathbb{Z}_{\geq 0}$. Since $\mathrm{LT}(a)\,\mathrm{LT}(b) = \mathrm{LT}(ab)$ for any monomial ordering, the basis obtained by multiplying the generating

functions is also a Groebner one with respect to lexicographic ordering, and the corresponding semi-infinite matrix $Q(x)$ consists of triangular blocks. Clearly, in this case the order of coefficients in the basis generating function does matter. Furthermore, it is possible to compute the product of generating functions modulo $z^{\rho+1}$, since higher order terms cannot belong to the target module of IPs.

The described calculations lead to a GB of the module of IPs with respect to lexicographic ordering. In order to obtain the minimum weighted degree IP one has to transform it into a reduced GB with respect to any $(1, k)$-weighted ordering. This can be done using the algorithm described in [10].

The proposed algorithm was implemented in C++ programming language. Computer simulations show that for $(32, 17)$ RS code the divide-and-conquer method provides approximately 2.5 times reduction of the complexity compared to straightforward implementation of iterative interpolation algorithm [2]. Furthermore, it allows to implement some steps of the list decoding algorithm in parallel. Analytical estimation of the proposed algorithm complexity is the subject of future work.

4 Conclusions

In this paper a novel method for computing the product of zero-dimensional coprime ideals was described. It reduces the ideal multiplication problem to computing a linear convolution of two multivariate polynomials. To the best of author knowledge, the problem of fast ideal multiplication was not studied before and the proposed method seems to be the first one. The proposed method was employed in the divide-and-conquer RS list decoding algorithm. It was shown that employing the lexicographic monomial ordering allows one to simplify the post-processing step of this algorithm. The proposed method can be used also for algebraic soft-decision decoding of RS codes.

The method considered in this paper still allows many improvements. First, construction of GB introduces artificial term orderings, which may be excessively strong in the context of bivariate interpolation problem. Second, only one element of the GB is used in the factorization step of GS algorithm, i.e. a lot of unnecessary data is computed. Detailed analysis of these issues is the subject of further work.

References

[1] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1757–1767, September 1999.

[2] R. R. Nielsen and T. Hoholdt, "Decoding Reed-Solomon codes beyond half the minimum distance," in *Proceedings of the International Conference on Coding Theory and Cryptography*. Springer-Verlag, 1998.

[3] F. Parvaresh and A. Vardy, "Polynomial matrix-chain interpolation in Sudan-type Reed-Solomon decoders," in *Proceedings of IEEE ISIT*, 2004, p. 386.

[4] R. Koetter, J. Ma, A. Vardy, and A. Ahmed, "Efficient interpolation and factorization in algebraic soft-decision decoding of Reed-Solomon codes," in *Proceedings of IEEE ISIT*, 2003, p. 365.

[5] J. Ma, P. Trifonov, and A. Vardy, "Divide-and-conquer interpolation for list decoding of Reed-Solomon codes," in *Proceedings of IEEE ISIT*, 2004, p. 386.

[6] R. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 246–257, 2000.

[7] D. Cox, G. Little, and D. O'Shea, *Ideals, varieties and algorithms*. Springer-Verlag, 1992.

[8] T. Sauer, "Polynomial interpolation of minimal degree and Grobner bases," in *Groebner Bases and Applications* , ser. London Mathematical Society Lecture Notes, B. Buchberger and F. Winkler, Eds., vol. 251. Cambridge University Press, 1998, pp. 483–494.

[9] M. G. Marinari, H. M. Moller, and T. Mora, "Grobner bases of ideals defined by functionals with an application to ideals of projective points," *Applicable Algebra in Engineering, Communication and Computing*, vol. 4, pp. 103–145, 1993.

[10] A. Basiri and J.-C. Faugere, "Changing the ordering of Grobner bases with LLL: Case of two variables," in *Proceedings of the International symposium on Symbolic and algebraic computation*, 2003.