Boolean Circuit Complexity of Algebraic Interpolation Problems

Marek Karpinski *
Dept. of Computer Science
University of Bonn
and
International Computer Science Institute
Berkeley, California

Abstract. We present here some recent results on fast parallel interpolation of multivariate polynomials over finite fields. Some applications towards the general conversion algorithms for boolean functions are also formulated.

Introduction

We consider the general problem of interpolation of multivariate polynomials over finite fields given by black boxes (input oracles). In this setting we are given a polynomial f over $\mathrm{GF}[q]$, as a black box, and an information about its sparsity t (the bound on the number of nonzero coefficients). Given this, we must determine an extension $\mathrm{GF}[q^s]$ of $\mathrm{GF}[q]$, s as small as possible, and an efficient (deterministic boolean NC-algorithm, cf. [Co 85], [KR 88]) interpolation algorithm working over $\mathrm{GF}[q^s]$ to determine all coefficients of f in $\mathrm{GF}[q]$. Such a

*Supported in part by Leibniz Center for Research in Computer Science, by the DFG Grant KA 673/2-1, and by the SERC Grant GR-E 68297

general problem arises in a number of applications, e.g., in design of efficient algorithms in algebra, coding theory and combinatorial optimization (cf. [Ga 83], [Ga 84], [Ka 85], [KT 88], [MS 72], [GK 87], [BT 88], [GKS 88]). The interest in the parallel (boolean circuit) complexity of this problem has arisen recently in connection with the design of fast parallel algorithms for the perfect matching problem [GK 87]. [GK 87] gave the first deterministic algorithm for sparse interpolation of determinants over fields of characteristic 0, and [BT 88] extended it to the case of arbitrary sparse polynomials over fields of characteristic 0.

Following [GK 87], and [GKS 88] we shall use uniform boolean circuits in our analysis. Given a (fixed) finite field GF$[q]$. We say that the *black box* Interpolation Problem (over a finite field extension GF$[q^s]$) is in NC^k (cf. [Co 85], [KR 88]), if there exists a class of uniform $(ntq)^{O(1)}$-size and $O(\log^k(ntq))$-depth boolean circuits with oracle nodes S (*returning* values of a black box over the field extension GF$[q^s]$) computing for an arbitrary n-variate polynomial $f \in \mathrm{GF}[q][x_1, \ldots, x_n]$ all the nonzero coefficients and monomial vectors of f. The oracle S_f^s is defined by $S_f^s(x_1, \ldots, x_n, y)$ iff $f(x_1, \ldots, x_n) = y$ over GF$[q^s]$. If the *lifting* of a black box (given explicitly by a straight-line program, determinant, boolean circuit, etc.) from the field GF$[q]$ to the extension GF$[q^s]$, and the computation of the value $f(x_1, \ldots, x_n)$ over GF$[q^s]$ by a black box, are both in boolean NC (in P), then the explicit Interpolation Problem lies also in boolean NC (in P).

The reader is referred to [LN 83], [MS 72] for the basic algorithms for finite fields, and to [Co 85], [KR 88] for the basic models of parallel computation.

1. Lower Bounds

We shall state first a result on the number of queries necessary to interpolate a sparse polynomial $f \in \mathrm{GF}[q][x_1, \ldots, x_n]$ over GF$[q]$ (i.e., for the case of $s = 1$).

Theorem 1. ([CDGK 88]) *Given an arbitrary finite field* GF$[q]$ *and a t-sparse polynomial $f \in \mathrm{GF}[q][x_1, \ldots, x_n]$ given by a black box input oracle, any algorithm for testing whether $f \equiv 0$ requires $\Omega(n^{\log t})$ queries to the input oracle.*

For the important case of boolean functions (GF[2]) we are able to prove the tight lower and upper bounds $\Theta(n^{\log t})$ for the number of queries necessary to determine identity to zero of t-sparse polynomials $f \in \mathrm{GF}[2][x_1, \ldots, x_n]$.

An Algorithm for GF[2] ([CDGK 88])

Input: t-sparse polynomial $f \in \mathrm{GF}[2][x_1, \ldots, x_n]$ given by a black box input oracle;

Output: **Yes**, if $f \equiv 0$; **No**, if $f \not\equiv 0$.

Step 1: **For all** n-bit vectors $v \in \{0,1\}^n$, having at most $\lfloor \log_2 t \rfloor$ zeros compute the values $\alpha_v = f(v)$.

Step 2: Output **Yes** iff $\forall v\, [\, \alpha_v = 0\,]$.

The correctness of the above algorithm was proven in [CDGK 88]. We do not know whether the result generalizes for the arbitrary finite field GF[q]. We note that we deal here not only with interpolation of polynomials but arbitrary functions in their RSE-representation ([We 87]).

For arbitrary boolean functions $f : \{0,1\}^n \to \{0,1\}$ there exists exactly one {0,1}-vector $S = (S_A)_{A \subseteq \{1,\ldots,n\}}$ such that

$$f(x) = \bigoplus_{A \subseteq \{1,\ldots,n\}} S_A \wedge \bigwedge_{i \in A} x_i$$

for $\bigoplus$ boolean *XOR* and $\bigwedge$ boolean *AND*.

The size of the vector S is referred to as the size (RSE(f)) of f in its RSE-representation (cf. [We 87]) (and is in our framework *exactly* its sparsity over $\mathrm{GF}[2][x_1, \ldots, x_n]$).

Theorem 2. ([CDGK 88]) *Given an arbitrary boolean function f by the black box input oracle, there exists an algorithm for deciding over* GF[2] *whether $f \equiv 0$ using $O(n^{\log(RSE(f))})$ queries to the oracle. The algorithm is optimal with respect to the number of queries to the oracle over* GF[2] *taken by any (adaptive or non-adaptive) algorithm for this problem.*

The lower bounds of this Section proves the impossibility of polynomial time (and NC-) algorithms for the general sparse polynomial interpolation with input oracles over finite fields without proper field extensions. So the intriguing question arises whether we can do interpolation over finite fields at all - without going to the 'impossible' field extension $\mathrm{GF}[q^n]$ (where there are no effective deterministic procedures known even for finding primitive elements!).

In the next section we shall present suprising upper bounds on the Interpolation Problem using only '*slight*' (logarithmic in nt) extensions of a ground field.

2. Upper Bounds

We formulate now our main Interpolation result on the *slight* field extensions.

Theorem 3. **([GKS 88])** *Given any t-sparse polynomial $f \in \mathrm{GF}[q][x_1, \ldots, x_n]$ by the black box input oracle. There exists a deterministic parallel algorithm* (NC^3) *for interpolating f over a slight field extension* $\mathrm{GF}[q^{\lceil 2\log_q(nt)+3\rceil}]$ *working in $O(\log^3(ntq))$ parallel boolean time and $O(n^2t^6\log^2(ntq) + q^{2.5}\log^2 q)$ processors. For the fixed field* $\mathrm{GF}[q]$, *the algorithm works in $O(\log^3(nt))$ parallel boolean time and $O(n^2t^6\log^2(nt))$ processors.*

The algorithm discovered in [GKS 88] involves two major computational steps: (1) breaking the zero identity problem of polynomials over a slight field extension $\mathrm{GF}[q^{\lceil 2\log_q(nt)+3\rceil}]$, and (2) inductive enumeration of all partial solutions for terms and coefficient vectors over $\mathrm{GF}[q]$ by means of recursion using (1).

We develop here a new general method involving Cauchy ([C]) matrices to break zero-identity problem in Step 1, and combine it with the new parallel enumeration method based on [GK 87] to solve Step 2. The number of queries to the input oracle over the slight field extension $\mathrm{GF}[q^{\lceil 2\log_q(nt)+3\rceil}]$ is bounded by $t(1+(n-1)\binom{t}{2})$ $(= O(nt^3))$.

We shall investigate here in more detail the problem of checking identity to zero (Step 1) in order to compare it with the results of Section 1. (The method of Cauchy matrices applied here could be also of independent interest.)

Definition. (Cauchy matrix) ([C])
An $(N \times N)$ matrix $C = [c_{ij}]$ over the field GF$[q]$ is called a Cauchy matrix, if

$$c_{ij} = \frac{1}{x_i + y_j}$$

for the fixed values $x_i, y_j \in \mathrm{GF}[q]$, $1 \leq i, j \leq N$.

Lemma. (cf., e.g. [MS 72]). *Let C be a Cauchy matrix, then the determinant*

$$Det(C) = \frac{\prod_{1 \leq i < j \leq N} (x_j - x_i)(y_j - y_i)}{\prod_{1 \leq i,j \leq N} (x_i + y_j)}$$

For any of its minors $\neq 0$ a similar formula holds. Therefore any minor of any size is nonsingular.

In our algorithm we construct the Cauchy matrix $C = [c_{ij}]$ by

$$c_{ij} = \frac{1}{i + j} \bmod p$$

where p is a prime.

An Algorithm for (the slight field extension) GF$[q^{O(\log(nt))}]$ ([GKS 88])

Input: t-sparse polynomial $f \in \mathrm{GF}[q][x_1, \ldots, x_n]$ given by a black box input oracle;

Output: **Yes**, if $f \equiv 0$; **No**, if $f \not\equiv 0$.

Step 1: Determine a minimal s satisfying

$$q^s - 1 > 4nq(n-1)\binom{t}{2}.$$

So take $s = \lceil \log_q(nt) + 3 \rceil$.

Step 2: Construct the field GF$[q^s]$ and a primitive element ω in GF$[q^s]$ with the help of Berlekamp Algorithm [Be 70].

Step 3: Let $N = \frac{\lceil q^s - 1 \rceil}{4nq}$. Use the sieve of Erastosthenes to find a prime p with $2N < p \leq 4N$.

Step 4: Construct an $N \times N$ Cauchy matrix $C = [c_{ij}]$ by $c_{ij} = \frac{1}{i+j} \bmod p, 1 \leq i, j \leq N$ by means of the Euclidean algorithm.

Step 5: Construct an arbitrary submatrix $\bar{C} = [\bar{c}_{ij}]$ of C of size $N \times n$.

Step 6: Query in parallel the black box for any row $\bar{c}_i = (\bar{c}_{ij})$, $1 \leq j \leq n$, of the matrix $\bar{C}$, and for each l, $0 \leq l < t$, at the points

$$\alpha_{l_i} = \omega^{l \cdot \bar{c}_i} = (\omega^{l \cdot \bar{c}_{i1}}, \omega^{l \cdot \bar{c}_{i2}}, \ldots, \omega^{l \cdot \bar{c}_{in}})$$

and at the zero point $\alpha_{0_0} = (0, \ldots, 0)$.

Step 7: Output **Yes** $(f \equiv 0)$ iff $\forall 0 \leq l < t, 0 \leq i \leq N \, [\, \alpha_{l_i} = 0 \,]$.

The correctness proof of this algorithm is given in [GKS 88]. The main reason for the algorithm to work is the strong *term separation* property of a Cauchy matrix constructed in Step 3 (the existence of a row $\bar{c}_i$ of a Cauchy matrix C separating arbitrary two monomials of f under $\omega^{l \cdot \bar{c}_i}$ substitution to the black box oracle).

We summarize

Theorem 4. *Given any fixed finite field* GF$[q]$ *and a t-sparse polynomial $f \in \mathrm{GF}[q][x_1, \ldots, x_n]$ by the black box input oracle, there exists a deterministic parallel algorithm* (NC^2) *for interpolating f over* GF$[q^{\lceil 2 \log_q(nt)+3 \rceil}]$ *working in $O(\log^2(nt))$ parallel boolean time and $O(n^2t^3)$ processors, and making $O(nt^3)$ queries to the black box input oracle.*

Theorems 3 and 4 can be generalized to work over arbitrary fields of positive characteristic, by applying our method to the *slight* extensions of their primitive subfields of the same characteristic.

3. Some Consequences for Boolean Function

We shall derive some interesting consequences of Theorem 3 and 4 for the case of boolean functions (GF[2]). Although we formulate them for GF[2] only, same result holds for arbitrary 'small' (or fixed) finite fields (in this case instead of boolean circuits we use straight-line programs!).

The boolean RSE-Conversion Problem is the problem of converting a boolean function f (given by the input oracle), and such that RSE$(f) \leq t$ into the equivalent RSE-formula.

A SPARSE$_\oplus$–SAT problem is the problem of checking whether f (given as above) has a satisfying assignment.

Theorem 3 entails directly the following.

Corollary 1. *The boolean* RSE-Conversion Problem *is in* NC^3. *The algorithm uses* $O(\log^3(nt))$ *parallel boolean time and* $O(n^2 t^6 \log^2(nt))$ *processors.*

It is interesting to note that $\mathrm{SPARSE}_{\oplus}$–SAT problem was not known before to be in P. Theorem 1 says that there is no polynomial time algorithm without using proper field extensions. Corollary 1 puts this problem in P and deterministic boolean NC^3, and Theorem 4 yields even better $O(\log^2 n)$ parallel time bound.

Corollary 2. $\mathrm{SPARSE}_{\oplus}$–SAT *is in* NC^2. *The algorithm uses* $O(\log^2(nt))$ *parallel boolean time and* $O(n^2 t^3)$ *processors.*

4. Further Research

The research on parallel complexity of multivariate polynomial interpolation was spurred by its application towards the parallel matching algorithms (cf. [GK 87]), and resulted already in several applications in problems like sparse factorization and polynomial GCD (cf. [KT 88], [BT 88]). The good bit-complexity algorithms require however computations over finite fields rather than $\mathbb{Z}$. In this connection an important problem arises to improve on the number of processors of the algorithms of Theorem 3, and 4.

References

[AL 86] Adleman, L.M., Lenstra, H.K., *Finding Irreducible Polynomials over Finite Fields*, Proc. 18^{th} ACM STOC (1986), pp. 350–355.

[B 81] Ben-Or, M., *Probabilistic Algorithms in Finite Fields*, Proc. 22^{th} IEEE FOCS (1981), pp. 394–398.

[BT 88] Ben-Or, M., Tiwari, P., *A Deterministic Algorithm for Sparse Multivariate Polynomial Interpolation*, Proc. 20^{th} ACM STOC (1988), pp. 301–309.

[Be 70] Berlekamp, E.R., *Factoring Polynomials over Large Finite Fields*, Math. Comp. 24 (1970), pp. 713–753.

[C] Cauchy, A.L., *Exercises d'Analyse et de Phys. Math.*, Vol 2, Paris, Bachelier (1841), pp.151–159.

[CDGK 88] Clausen, M., Dress, A., Grabmeier, J., Karpinski, M., *On Zero-Testing and Interpolation of k-Sparse Multivariate Polynomials over Finite Fields*, Research Report No.8522-CS, University of Bonn (1988); to appear in Theoretical Computer Science (1989).

[Co 85] Cook, S.A., *A Taxonomy of Problems with Fast Parallel Algorithms*, Information and Control 64 (1985), pp. 2–22.

[Ga 83] von zur Gathen, J., *Factoring Sparse Multivariate Polynomials*, Proc. 24^{th} IEEE FOCS (1983), pp. 172–179.

[Ga 84] von zur Gathen, J., *Parallel Algorithm for Algebraic Problems*, SIAM J. Comput., 13 (1984), 808–824.

[GK 87] Grigoriev, D.Y., Karpinski, M., *The Matching Problem for Bipartite Graphs with Polynomially Bounded Permanents is in NC*, Proc. 28^{th} IEEE FOCS (1987), Los Angeles, 1987, pp. 166–172.

[GKS 88] Grigoriev, D.Y., Karpinski, M., Singer, M.F., *Fast Parallel Algorithms for Sparse Multivariate Polynomial Interpolation over Finite Fields*, Research Report No.8523-CS, University of Bonn (1988); submitted to SIAM J. Comput., 1988.

[Ka 85] Kaltofen, E., *Computing with Polynomials Given by Straight-Line Programs I, Greatest Common Divisors*, Proc. 17^{th} ACM STOC (1985), pp. 131–142.

[KT 88] Kaltofen, E., Trager, B., *Computing with Polynomials Given by Black Boxes for their Evaluations: Greatest Common Divisor, Factorization, Seperation of Numerators and Denominators*, Proc. 29^{th} IEEE FOCS (1988), pp. 296–305.

[KR 88] Karp, R.M., Remachandran,V., *A Survey of Parallel Algorithms for Shared-Memory Machines*, Research Report No.UCB/CSD 88/407, University of California, Berkeley (1988); to appear in Handbook of Theoretical Computer Science, North Holland (1989).

[LN 83] Lidl, H., Niederreiter, H., *Finite Fields, Encyclopedia of Mathematics and its Applications*, Vol.10, Cambridge University Press (1983).

[MS 72] MacWilliams, F.J., Sloane, N.J.A., *The Theory of Error Correcting Codes*, North Holland (1972).

[We 87] Wegener, I., *The Complexity of Boolean Functions*, Teubner (1987).