A Multi-level Blocking Distinct Degree Factorization Algorithm

Richard P. Brent
Australian National University
Canberra, Australia
Fq8@rpbrent.com

Paul Zimmermann
INRIA Lorraine/LORIA
Villers-lès-Nancy, France
Paul.Zimmermann@loria.fr

Abstract

We give a new algorithm for performing the distinct-degree factorization of a polynomial $P(x)$ over GF(2). Our search algorithm uses a multi-level blocking strategy. The coarsest level of blocking replaces GCD computations by multiplications, as suggested by Pollard [*BIT* 15 (1975), 331–334], von zur Gathen and Shoup [*Computational Complexity* 2 (1992), 187–224], and others.

The novelty of our approach is that a finer level of blocking replaces multiplications by squarings, which speeds up the computation in $\mathrm{GF}(2)[x]/P(x)$ of the interval polynomials $p_m(x^{2^d}, x)$, where

$$p_m(X, x) = \prod_{j=0}^{m-1} \left(X^{2^j} + x \right) = \sum_{j=0}^{m} x^{m-j} s_{j,m}(X) \ , \ \ s_{j,m}(X) = \sum_{0 \le k < 2^m, \ w(k)=j} X^k \ ,$$

and $w(k)$ denotes the Hamming weight of k. Now $p_m(x^{2^d}, x)$ can be computed with cost $m^2 S(r)$ if we already know $s_{j,m}(x^{2^{d-m}})$ for $0 \le j \le m$. Here $S(r)$ is the cost of a squaring in $\mathrm{GF}(2)[x]/P(x)$, and r is the degree of $P(x)$. If $P(x)$ is a trinomial then $S(r) = \Theta(r)$, which we assume below, although the algorithm applies more generally.

Multiplication of polynomials of degree r over GF(2) can be performed in time $M(r) = O(r \log r \log \log r)$. We have implemented an algorithm of Schönhage [*Acta Inf.* 7 (1977), 395–398] that achieves this bound. We also consider the classical, Karatsuba and Toom-Cook algorithms that have $M(r) = O(r^\alpha)$, $1 < \alpha \le 2$, since these algorithms are easier to implement and are faster for small r.

The optimal value of m satisfies $m^2 S(r) \approx M(r)$, so the optimal m is $\Theta(\sqrt{M(r)/r})$ and we gain a speedup $\sim m/2$ over the classical single-level blocking algorithm (the case $m = 1$).

As an application we give a fast algorithm to search for all irreducible trinomials $x^r + x^s + 1$ of degree r over GF(2), while producing a certificate that can be checked in less time than the full search. The certificate is simply a list giving, for each trinomial that is reducible, a factor of minimal degree. Classical algorithms cost $O(r^2)$ per trinomial, thus $O(r^3)$ to search over all trinomials of given degree r. (This can be reduced to $O(r^3/\log r)$ if the "easy" cases with a factor of degree less than $\log_2 r$ are handled efficiently.)

Under a plausible assumption about the distribution of factors of trinomials, our algorithm has complexity $O(r^2 \log r \sqrt{M(r)/r}) = O(r^2 (\log r)^{3/2} (\log \log r)^{1/2})$ for the search over all trinomials of degree r. Verification can be performed in time $O(nr^2) + \widetilde{O}(r)$, where n is the number of irreducible trinomials found.

Our implementation achieves a speedup of greater than a factor of 70 over the classical algorithm in the case $r = 6972593$ considered by Brent, Larvala and Zimmermann [*Math. Comp.* 74, (2005), 1001–1002], so there is a good prospect of finding irreducible trinomials of even larger degree.

In the paper we discuss the multi-level blocking strategy in detail, and describe a back-tracking strategy to handle the case where more than one irreducible factor is found in a large block.