Accelerating the Factorization of Multilinear Boolean Polynomials

Tian Chen and Michael Monagan
tca71@sfu.ca, mmonagan@sfu.ca
Department of Mathematics, Simon Fraser University, Canada

Abstract

A multilinear Boolean polynomial is a polynomial over $GF(2)$ in which each variable has degree at most 1 and every coefficient is 0 or 1. Such polynomials play a key role in applications including Boolean circuit optimization, yet their efficient factorization remains challenging. We present two different algorithms that advance this problem.

First, we introduce a Monte Carlo factorization algorithm with algebraic complexity $O(n^2t)$ over a suitable extension field $GF(2^k)$, where n is the number of variables and t is the number of terms of the input polynomial. Our C implementation achieves substantial speedups over both the FDE and GCD algorithms of Emelyanov–Ponomaryov [3]. In this implementation, the dominant bottleneck is multiplication in $GF(2^{63})$; we address this using a compact 64-bit representation together with an optimized bit-shift/XOR reduction routine.

On the other hand, when the input polynomial can also be represented by a black box, we apply our recently developed black-box factorization algorithm CMBBSHL [1, 2], implemented in Maple and C. The black-box representation allows us to reduce the effective parameter t (the number of terms of the input polynomial) to $s_{\max} \sum \#f_i + C(\text{probe})$ (typically $\ll t$), where $s_{\max}$ is the maximum number of terms in any coefficient of any factor in x_1, $\sum \#f_i$ is the sum of the number of terms of all factors, and $C(\text{probe})$ is the cost of a single black-box probe. This yields an overall algebraic complexity $O\left(n^2\left(s_{\max} \sum \#f_i + C(\text{probe})\right)\right)$. The dominant cost in practice is usually probing the black box. To mitigate this, we implement fast C routines for matrix construction and determinant computation in $\mathbb{Z}_p$, used when the black box computes symbolic determinants. In the less common case where some factor has a very large number of terms, the main cost becomes solving Vandermonde systems; for this we implemented a fast Vandermonde solver in Maple.

Together, these methods, supported by efficient Maple and C implementations, provide the fastest practical toolkit to date for factoring multilinear Boolean polynomials.

References

[1] T. Chen and M. Monagan. A new black box factorization algorithm – the non-monic case. In Proceedings of ISSAC 2023. ACM, 2023

[2] T. Chen. Sparse Hensel lifting algorithms for multivariate polynomial factorization. PhD Thesis (2024)

[3] Pavel Emelyanov and Denis Ponomaryov. On a Polytime Factorization Algorithm for Mul- tilinear Polynomials over $\mathbb{F}_2$. Proceedings of CASC 2018, LLNCS 11077:164-176, Springer, 2018.

[4] Jiaxiong Hu and Michael Monagan. A Fast Parallel Sparse Polynomial GCD Algorithm. J. Symb. Cmpt. 105:(1) 28-63, Springer, July 2021.

[5] Michael Monagan and Roman Pearce. The design of Maple's sum-of-products and POLY data structures for representing mathematical objects. Communications in Computer Algebra, 48(4):166-186, ACM, December 2014.

[6] Amir Shpilka and Ilya Volkovich. On the relation between polynomial identity testing and finding variable disjoint factors. Proceedings of ICALP 2010. LNCS 6198:408-419, Springer, 2010.

[7] Richard Zippel. Probabilistic algorithms for sparse polynomials. Proceedings of EUROSAM '79, LNCS 72:216-226, Springer, 1979.