Complexity of Factorization in Products of Linear Forms

Pascal Koiran, LIP, ENS Lyon

October 2018

In a recent preprint [7] we have proposed three algorithms for the factorization of multivariate polynomials into products of linear forms. Our algorithms work in the black box model, where the algorithm has access to the polynomial f to be factored only through a "black box" which on input $(x_1, \ldots, x_n)$ outputs $f(x_1, \ldots, x_n)$. The problem of polynomial factorization in the black box model was solved in full generality by Kaltofen and Trager [2]. Our main goal in [7] was to give simpler algorithms for the special case of factorization into products of linear forms. In this internship several directions for improving our algorithms will be explored. In particular, our first algorithm seems to be the only factorization algorithm based on ideas from invariant theory.[1] This should be a fruitful direction to pursue since it is as of now unexplored in the context of factorization algorithms, except of course for [7]. We propose below three possible research directions.

1. Improved running time: following ideas from [6], our invariant-theoretic algorithm first determines the Lie algebra of the input polynomial f. This boils down to the resolution of a linear system [5, 6], but this step is nonetheless expensive since the system to be solved has n^2 unknowns. One could try to exploit the special structure of the linear system to speed up its resolution.

2. Randomization: our three algorithms are randomized. As pointed out in [7], this is unavoidable for polynomial time algorithms in the black box model. Nonetheless, looking for a deterministic algorithm makes sense in the model where the input polynomial is given by an arithmetic circuit rather than a black box. This question even makes sense for the black box model if we assume that the input can be factorized as a product of linear forms (in this case, the algorithm must output such a factorization but it does not have to decide whether such a factorization is possible). In order to obtain a deterministic factorization algorithm, one could try to derandomize the computation of the Lie algebra of

[1] A polynomial f is invariant under a linear transformation A if $f(A.x) = f(x)$ for all x.

f. This would be of interest beyond polynomial factorization since the computation of the Lie algebra has other applications [6]. Note also that obtaining a deterministic algorithm for the general problem of polynomial factorization is equivalent to the notorious open problem of derandomizing polynomial identity testing (PIT) [8].

3. Large exponents: an arithmetic circuit of size s can compute a polynomial of degree up to 2^s. The factorization of such high-degree polynomials raises new issues compared to the low degree case. For instance, it is a remarkable fact that the factors of an arithmetic circuit of "low" degree can all be computed by "small" arithmetic circuits [4, 3] (see also Theorems 2.21 and 8.14 in [1]); but this is not known for the low-degree factors of high-degree circuits ([1], Conjecture 8.3). In [7] we studied factorization into products of linear forms for the low degree case only. The case of high degrees (i.e., large exponents for the linear forms) could be investigated in this internship.

Student's background. The student should be interested in algorithms and complexity, and have some prior exposure to these subjects. He or she should be comfortable working with polynomials. Note however that knowledge of advanced topics in algebra (and in particular Lie algebras) is *not* a prerequisite since the required notions are presented in an elementary and self-contained way in [6, 7].

In this internship the student will have the opportunity to design and analyze new algorithms; and will get exposure to current research topics such as derandomization of algorithms or invariant-theoretic methods in algebraic complexity [5, 6, 7].

References

[1] P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory.* Number 7 in Algorithms and Computation in Mathematics. Springer, 2000.

[2] E. Kaltofen and B. Trager. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *Journal of Symbolic Computation*, 9(3):301–320, 1990.

[3] Erich Kaltofen. Single-factor Hensel lifting and its application to the straight-line complexity of certain polynomials. In *Proc. 19th ACM Symposium on Theory of Computing (STOC)*, pages 443–452, 1986.

[4] Erich Kaltofen. Factorization of polynomials given by straight-line programs. In *Randomness and Computation*, pages 375–412. JAI Press, 1989.

[5] Neeraj Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In *Symposium on Discrete Algorithms (SODA)*. Society for Industrial and Applied Mathematics, January 2011.

[6] Neeraj Kayal. Affine projections of polynomials. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC)*, pages 643–662, 2012.

[7] P. Koiran and N. Ressayre. Orbits of monomials and factorization into products of linear forms. *arXiv:1807.03663*, 2018.

[8] Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of polynomial identity testing and deterministic multivariate polynomial factorization. In *Proc. 29th Conference on Computational Complexity (CCC)*, pages 169–180, 2014.