PhD position in computer science at LIRMM (Montpellier), France

Informations

Location: Montpellier, France at LIRMM (www.lirmm.fr)

Duration: 36 months

Starting date: October 2019

Supervisors:

- Pascal Giorgi - pascal.giorgi@lirmm.fr
- Bruno Grenet - bruno.grenet@lirmm.fr

Qualifications

The candidate must hold a MSc (Master) in either computer science or computational mathematics. He/She must have good knowledge in complexity theory and/or computer algebra.

PhD proposal

Keywords: computer algebra, linear algebra, polynomial arithmetic, complexity theory, sparse polynomials, arithmetic circuits

Title: Algorithms for Sparse Polynomials: from lower bounds to practical applications

Polynomial arithmetic is a fundamental tool in both computer algebra and complexity theory with applications in many domains from cryptography to combinatorics. The dense representation of polynomials have been widely studied in the litterature: in particular following the major breakthrough result in the 60's by Cooley and Tuckey on Fast Fourier Transform [2] which makes it possible to multiply dense polynomials in quasi-linear time. In the sparse representation case much fewer results are known while it is a more efficient way to represent such mathematical objects, especially in the multivariate setting. For instance, no similar result of quasi-linear time algorithm exists for the multiplication in this case [9]. Besides, the study of such structured polynomials is at the core of the algebraic complexity theory defined by Valiant in [11] as an approach for the P *vs* NP problem.

The central goal of this thesis is to provide a better understanding of sparse polynomial arithmetic, ranging from very simple tasks such as multiplication or division to more involved questions about the factorisation of polynomials given in sparse representation. The proposed approach is to mix mathematical investigations to provide new algorithms and lower bounds on the problems and practical efficient implementations. The two facets are interrelated and shall feed each other.

The basics of sparse polynomial arithmetic, such as multiplication, division and gcd computation, are not yet fully understood [9]. For division or gcd computation, it is not even known whether polynomial time algorithms exist in general. The role of cyclotomic polynomials in these questions have to be investigated further. One possible solution is to introduce cyclotomic polynomials in the representation [3]. Another way of enriching the representation is to use the formalism of arithmetic circuits that originates from algebraic complexity. In particular, sparse polynomials can be viewed as circuit of depth 2. Allowing more general constant depth circuits in the representation may be a step towards faster algorithms. The questions raised by this approach are also of interest for the algebraic complexity community.

Factorisation problems are much more involved than multiplication or division, and generic results might be even harder to obtain. In particular, the irreducible factorisation of a sparse polynomial may require superpolynomial size. A possible workaround consists in computing only a part of the factorization such as low-degree factors [8]. We wish to extend this idea to *small* factors more generally, that is computing those factors that do not require a large representation. Another avenue worth exploring is the use of constant depth circuits again: This is deeply connected to structural questions investigated by the algebraic complexity community [5] and could be a way to overcome the current barriers on sparse polynomial factorisation.

Another promising line of research on the subject would be the definition of deterministic or probabilistic verification algorithms for sparse polynomial computations, the goal being that verification costs less than the

computation itself. This kind of procedure is of great interest for outsourced computations in the cloud. While it is known how to probabilistically verify that a dense polynomial product is correct (simply evaluate the operands and the result at random points and check equality [12, 4, 10]), it is not yet known how to provide similar results in the sparse case. Indeed, the degree might be exponentially larger than the number of terms and evaluation is therefore prohibited. In this context, some generic ideas coming from complexity theory on provable circuits [7] can be investigated, or the use of some linear algebra techniques on a dedicated problem as in [6] can be explored.

More precisely, the candidate could investigate one of the following tasks as a starting point for the thesis:

- Certification of sparse multiplication can be handled through the certification of dense modular multiplication. Indeed, the idea is to map the initial problem modulo a smallish dense polynomial and then to perform the equality test within the dense polynomial setting. There, the remaining problem is to provide a linear time verification for dense polynomial modular multiplication and provide a satisfying probabilistic analysis.
- The division of sparse polynomials has been less investigated than multiplication. There, the complexity result on multiplication is almost optimal in the output size while it is not true for division. One approach is to find tools from fast multiplication algorithms that are applicable to division.
- An easier variant of gcd computation is when the two input polynomials are assumed cyclotomic free. There, the open question is whether it exists a polynomial algorithm to solve this problem. One approach to derive such an algorithm would be to exploit mathematical results on cyclotomic free sparse polynomials given in [1].

References

[1] Francesco Amoroso, Louis Leroux, and Martín Sombra. Overdetermined systems of sparse polynomial equations. *Foundations of Computational Mathematics*, 15(1):53–87, Feb 2015.

[2] James Cooley and John Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[3] James H. Davenport and Jacques Carette. The Sparsity Challenges. In Stephen M. Watt, Viorel Negru, Tetsuo Ida, Tudor Jebelean, Dana Petcu, and Daniela Zaharie, editors, *11th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2009, Timisoara, Romania, September 26-29, 2009*, pages 3–7. IEEE Computer Society, 2009.

[4] Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193 – 195, 1978.

[5] Michael A. Forbes and Amir Shpilka. Complexity Theory Column 88: Challenges in Polynomial Factorization. *ACM SIGACT News*, 46(4):32–49, 2015.

[6] Pascal Giorgi. A probabilistic algorithm for verifying polynomial middle product in linear time. *Information Processing Letters*, 139:30–34, 2018.

[7] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 113–122, New York, NY, USA, 2008. ACM.

[8] Bruno Grenet. Bounded-degree factors of lacunary multivariate polynomials. *Journal of Symbolic Computation*, 75:171–192, July 2016.

[9] Daniel S. Roche. What can (and can't) we do with sparse polynomials? In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, ISSAC '18, pages 25–30, New York, NY, USA, 2018. ACM.

[10] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.

[11] Leslie G. Valiant. Completeness classes in algebra. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 249–261, New York, NY, USA, 1979. ACM.

[12] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation. In: Lecture Notes in Comput. Sci., vol. 72*, pages 216–226. Springer-Verlag, 1979.