ANALYSIS OF FAST VERSIONS OF THE EUCLID ALGORITHM

LOICK LHOTE

Joint work with
Eda Cesaratto (Facultad de Ingeniera, Universidad de Buenos Aires, Argentina, and GREYC, Université de Caen), Julien Clément (GREYC, Université de Caen), Benoît Daireaux(IrisResearch Center, Stavanger, Norway), Véronique Maume-Deschamps(IMB, Université de Bourgogne), Brigitte Vallée (GREYC, Université de Caen)

This work has been already presented in the ANALCO'07 conference, and a larger abstract already appeared in the proceedings of this conference [5]. *A long paper on this subject is in preparation.*

1. Introduction

Gcd computation is a widely used routine in computations on long integers. It is omnipresent in rational computations, public key cryptography or computer algebra. Many gcd algorithms have been designed since Euclid. Most of them compute a sequence of remainders by successive divisions, which leads to algorithms with a quadratic bit–complexity (in the worst-case and in the average-case). Using Lehmer's ideas [11] (which replace large divisions by large multiplications and small divisions), computations can be sped-up by a constant factor, but the asymptotic complexity remains quadratic. Major improvements in this area are due to Knuth [10], who designed the first subquadratic algorithm in 1970, and to Schönhage [14] who subsequently improved it the same year. They use both Lehmer's ideas and Divide and Conquer techniques to compute in a recursive way the quotient sequence (whose total size is $O(n)$). Moreover, if a fast multiplication with subquadratic complexity (FFT, Karatsuba...) is performed, then one obtains a subquadratic gcd algorithm (in the worst-case). Such a methodology has been recently used by Stehlé and Zimmermann [15] to design a Least-Significant-Bit version of the Knuth-Schönhage algorithm. According to experiments due to [4] or [13], these algorithms (with a FFT multiplication) become efficient only for integers of size larger than 10000 words, whereas, with the Karatsuba multiplication, they are efficient for smaller integers (around 100 words). A precise description of the Knuth-Schönhage algorithm can be found in [19, 13] for instance.

2. Previous results on gcd algorithms

The average-case behaviour of the quadratic gcd algorithms is now well understood. First results are due to Heilbronn and Dixon in the seventies, who studied for the first time the mean number of iterations of the Euclid Algorithm, then Brent analysed the Binary algorithm [3], and Hensley [9] provided the first distributional analysis for the number of steps of the Euclid Algorithm. Since 90, the Caen Group [16, 18, 17] has performed an average-case analysis of various parameters of a large class of Euclidean algorithms. More recently, distributional results have also been obtained for the Euclid algorithm and some of its variants: first Baladi and Vallée prove that a whole class of so–called additive costs of moderate growth follows an asymptotic gaussian law [2] (for instance, the number of iterations, the number of occurrences of a given digit, etc). This year, Lhote and Vallée [12] show that a more general class of parameters also follows an asymptotic gaussian law. This class contains the length of a remainder at a fraction of the execution, and the bit-complexity.

To the best of our knowledge, there are yet few results on "efficient algorithms". In [7], the authors perform an average-case analysis of Lehmer's algorithm, and exhibit the average speed-up obtained using these techniques. As far as we know, there does not any probabilistic analysis on subquadratic algorithms. This is the goal of the paper to present such a study with variants of the Knuth-Schönhage algorithm.

3. Analysis of Knuth-Schönhage algorithms

There are two algorithms to be analyzed, the $\mathcal{HG}$ algorithm and the $\mathcal{G}$ algorithm. The $\mathcal{G}$ algorithm computes the gcd and is composed by successive calls to the $\mathcal{HG}$ algorithm. The $\mathcal{HG}$ algorithm (for Half-Gcd Algorithm) only returns the same result as the the "first half" of the classical Euclid algorithm. It admits a Divide and Conquer form with two recursive calls to itself and some arithmetic computations between and after the recursive calls.

Interrupted Euclidean algorithms. We first show that these algorithms can be viewed as a sequence of the so–called Interrupted Euclidean algorithms. An Interrupted Euclidean algorithm is a subsequence formed by successive iterations of the classical Euclid algorithm. On an input (A, B), the classical Euclid algorithm builds a sequence of remainders A_i, a sequence of quotients Q_i, and a sequence of matrix $\mathcal{M}_i$. On an input (A, B) of binary size n, the Interrupted Euclidean algorithm $\mathcal{E}_{[\delta,\delta+\gamma]}$ starts at the index k of the execution of the Euclid Algorithm, as soon as the remainder A_k has already lost $\delta\, n$ bits (with respect to the initial A which has n bits) and stops at index $k+i$ as soon as the remainder A_{k+i} has lost $\gamma\, n$ supplementary bits (with respect to the remainder A_k). The $\mathcal{HG}$ algorithm is just an algorithm which simulates the interrupted algorithm $\mathcal{E}_{[0,1/2]}$. A quite natural question is: How many iterations are necessary to lose these $\gamma\, n$ bits? Of course, it is natural to expect that this subsequence of the Euclidean algorithm is just "regular" and locally similar to the "total" Euclidean Algorithm; in this case, the number of iterations would be close to $\gamma\, P$ (where P is the number of iterations of the "total" Euclid algorithm). Our first result describes precisely this regularity. Indeed, on the set Ω_n formed with rational inputs of size n endowed with an initial density f smooth enough, the random variable P_δ equal to the number of iterations performed by the interrupted algorithm $\mathcal{E}_{[0,\delta]}$ satisfies a gaussian limit law with asymptotic mean,

$$\mathbb{E}_{n,f}[P_\delta] \sim \delta \cdot \mathbb{E}_{n,f}[P],$$

for large entries of size n.

Evolution of the distribution. For a probabilistic study of $\mathcal{G}$ and $\mathcal{HG}$, a precise description of the evolution of the distribution during the execution of the classical Euclid Algorithm is of crucial interest. For real inputs, we know that the continued fraction algorithm does not terminate (except for rationals ...). Moreover, as the continued fraction algorithm is executed, the distribution of reals tends to the distribution relative to the Gauss density φ, defined as

$$\varphi(x) = \frac{1}{\log 2}\frac{1}{1+x}. \tag{1}$$

For rational inputs, one begins with some distribution f on the set of the inputs $x := A_1/A_0$ of size n, and we consider the rationals $x_k := A_{k+1}/A_k$. We focus on the first index k where the binary size of x_k is less than $(1-\delta)n$ and we denote the corresponding rational x_k by $x_{\langle\delta\rangle}$. What is the distribution of the rational $x_{\langle\delta\rangle}$? The evolution of this distribution is clearly more intricate than in the real case, since at the end of the algorithm (when $\delta = 1$), the distribution is the Dirac measure at $x = 0$. We obtain here a precise description of this distribution (Figure 1) which involves another density

$$\psi(x) := \frac{12}{\pi^2}\sum_{m\geq 1}\frac{\log(m+x)}{(m+x)(m+x+1)}. \tag{2}$$

In particular, we prove the following,

$$\mathbb{P}_{n,f}[x_{\langle\delta\rangle} \in J] = \left(\int_J \psi(t)dt\right)[1 + O\left(\beta_n(f,\delta,J)\right)].$$

Here, $\mathbb{P}_{n,f}$ is the probability over the entries of size n endowed with density f, J is any interval and the error term β_n is explicit and depends on the triple (f,δ,J).

Truncatures. In the original design of the algorithm, the inputs of the recursive calls are obtained from the initial inputs by using a truncation procedure. This procedure only keeps m dominant bits of the large integers (with n bits) in order to form small integers whose first quotients (in the execution of the classical Euclid algorithm) are the same as those computed on the large integers. Then, we also need precise results on the distribution of the truncatures of the remainders. Here, we deal with a probabilistic truncature, which is more regular than the original one used by Lehmer,

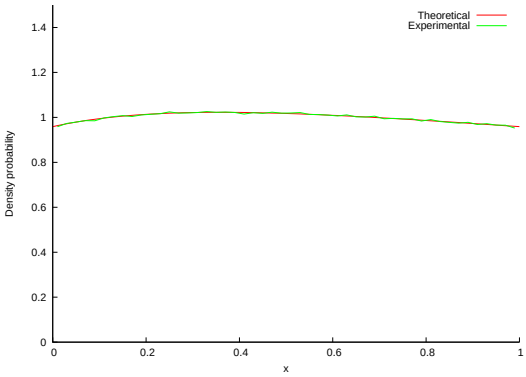

FIGURE 1. Density distribution of $x_{\langle\delta\rangle}$ in the case $\delta = 1/2$. This corresponds to the density distribution of the rational $x_k := A_{k+1}/A_k$ obtained as soon as $\ell(A_k)$ is smaller than $(1/2)\ell(A_0)$. The density distribution is plotted against its theoretical counterparts $\psi(x)$. Here we consider Ω_n for $n = 48$ (48 bits), the interval $[0, 1]$ is subdivided into equal subintervals of length $1/50$ to estimate the density, and 3 537 944 rational are drawn from Ω_n according to the initial density distribution φ by Monte Carlo simulation.

and we precisely compare the distribution of the truncated rationals with the distribution of the long rationals.

Finally, we are led to design precise variants of the classical algorithms, denoted by $\underline{\mathcal{HG}}$ and $\underline{\mathcal{G}}$, which take into account adequate choices of parameters in the fast algorithms and for which the precise analysis can be performed.

Adjust functions. The fast versions also deal with other functions, which are called the Adjust functions. Such functions perform few steps of the classical Euclid Algorithm. However, the bit-complexity of the Adjust functions depends on the size of the quotients which are computed during these steps. Even for estimating the worst–case complexity of the fast variants, the Adjust functions are not precisely analyzed. The usual argument is "The size of a quotient is $O(1)$". Of course, this assertion is false, since this is only true on average. Since the Adjust functions are related to some precise steps, the size of the quotients computed at these precise steps may be large. We are then led to study the probability that the Euclid Algorithm only produces "small" quotients. Precisely, Cesaratto and Vallée [6] prove that, on inputs of size n, and with a probability of the form $1 - O(n^{1-(\log\log n)^{1/2}})$, all the quotients are bounded by $\log n(\log\log n)^{1/2}$. And, we also prove this result for our truncated integers.

Finally, we obtain the exact average-case complexity of our versions of the two main algorithms of interest, the $\underline{\mathcal{HG}}$ algorithm, and the $\underline{\mathcal{G}}$ algorithm itself. We obtain the following results about the average bit-complexity B, G of both algorithms, on the set of random inputs of size n

$$\mathbb{E}_n[B] = \Theta(n(\log n)^2 \log\log n),$$
$$\mathbb{E}_n[G] = \Theta(n(\log n)^2 \log\log n).$$

Furthermore, we obtain some precise information about the constants which are involved in the Θ–terms. Then, our proven average bit–complexity of the $\underline{\mathcal{HG}}, \underline{\mathcal{G}}$ algorithms is of the same order as the usual heuristic bound on the worst-case complexity of $\mathcal{HG}, \mathcal{G}$ algorithms.

4. METHODS

All our main conclusions obtained here are "expected", and certainly do not surprise the reader. However, the irruption of the density ψ is unexpected, and an actual proof of this phenomenon is not straightforward. This is due to the fact that there is a correlation between successive steps of the Euclid Algorithm. Then, the tools which are usual in analysis of algorithms [8], as generating functions, are not sufficient to study this algorithm. All the analyses which will be described here are instances of the so–called dynamical analysis, where one proceeds in three main steps: First,

the (discrete) algorithm is extended into a continuous process, which can be defined in terms of the dynamical system related to the Gauss map. Then, the transfer operator $\mathbf{H}_s$ defined as

$$\mathbf{H}_s[f](x) := \sum_{m\geq 1} \frac{1}{(m+x)^{2s}} f\left(\frac{1}{m+x}\right)$$

explains how the distribution evolves, but only in the continuous world. The executions of the gcd algorithm are now described by particular trajectories (i.e., trajectories of "rational" points), and a transfer "from continuous to discrete" must finally be performed.

The present paper mainly uses two previous works, and can be viewed as an extension of them: first, the average-case analysis of the Lehmer-Euclid algorithm performed in [7], second the distributional methods described in [2, 12]. First, we again use the general framework that Daireaux and Vallée have defined for the analysis of the Lehmer-Euclid Algorithm, which explains how the Lehmer-Euclid algorithm can be viewed as a sequence of Interrupted Euclidean algorithms $\mathcal{E}_{[\delta,\delta+\gamma]}$. However, in [7], we only used some "easy" properties of the transfer operator $\mathbf{H}_s$. We now need proving that properties which were already crucial in previous distributional analysis [2, 1, 12] –namely, the *US* Property for the quasi-inverse $(I-\mathbf{H}_s)^{-1}$ of the transfer operator– also hold in our context. The *US* property can be summarized as follows:
For any ξ, there exists a vertical strip $\mathcal{S}$ of the form $|\Re s - 1| \leq \sigma$ for which the following holds:

(*i*) The quasi-inverse has a unique pôle in $\mathcal{S}$,
(*ii*) On the left line $\Re s = 1-\sigma$, one has $(I-\mathbf{H}_s)^{-1}[1] = O(|\Im s|^{\xi})$.

Here, our various theorems lead to study parameters of various type, whose generating functions involve various operators $\mathbb{G}_{s,w}$ which depend on two variables s, w. However, for small w's, all these operators can be viewed as a perturbation of the quasi-inverse $(I-\mathbf{H}_s)^{-1}$ and we have to prove that the *US* Property extends to these perturbated quasi-inverses. In particular, the existence of a strip $\mathcal{S}$ where the *US* property holds uniformly with respect to w is crucial in the analysis: the main choices of the parameters in our versions $\underline{\mathcal{HG}}$ and $\underline{\mathcal{G}}$ of the fast algorithms depend on the width σ of this strip.

References

[1] Baladi, V. and Vallée, B. Exponential Decay of Correlations for surface semi-flows without finite Markov partitions, Proceedings of the American Mathematical Society, 133 (3) pp 865-874, 2004.
[2] Baladi, V. and Vallée, B. Euclidean Algorithms are Gaussian, Journal of Number Theory, Volume 110, Issue 2 (2005) pp 331–386.
[3] Brent, R.P. Analysis of the Binary Euclidean algorithm, Algorithms and Complexity, New directions and recent results, ed. by J.F. Traub, Academic Press 1976, pp 321–355.
[4] Cesari, G. Parallel Implementation of Schönhage's Integer GCD Algorithm, Proceedings of ANTS-III, LNCS 1423, pp64-76.
[5] Cesaratto, E., Clément, J., Daireaux, B., Lhote, L., Maume-Deschamps, V. , Vallée, B. *Analysis of fast versions of the Euclid Algorithm*, Proceedings of ANALCO'07, Janvier 2007, 16 pages.
[6] Cesaratto, E. and Vallée, B. Personal communication, to be submitted.
[7] Daireaux, B., and Vallée, B. Dynamical analysis of the parameterized Lehmer-Euclid Algorithm, Combinatorics, Probability, Computing, pp 499–536 (2004).
[8] Flajolet, P. and Sedgewick, R. *Analytic Combinatorics*, Book in preparation (1999), see also INRIA Research Reports 1888, 2026, 2376, 2956.
[9] Hensley, D. The number of steps in the Euclidean algorithm, *Journal of Number Theory* 49, 2 (1994), 142–182.
[10] Knuth, D.E. The analysis of algorithms, Actes du Congrès des Mathématiciens, Volume 3, pp 269-274, Gauthier-Villars 1971.
[11] Lehmer, D. H. Euclid's algorithm for large numbers. Am. Math. Mon. (1938) 45 pp 227–233.
[12] Lhote, L. and Vallée, B. Sharp estimates for the main parameters of the Euclid Algorithm, Proceedings of LATIN'06, LNCS 3887, pp 689-702. Long paper *Gaussian laws for the main parameters of the Euclid Algorithms*, to appear in Algorithmica, 2007 [35 pages]
[13] Möller, N. On Schönhage's algorithm and subquadratic integer gcd computation, submitted.
[14] Schönhage, A. Schnelle Berechnung von Kettenbruchentwicklungen, Acta Informatica pp 139–144 (1971)
[15] Stehlé, D. and Zimmermann, P. A Binary Recursive Gcd Algorithm, Proceedings of ANTS'04, LNCS 3076 (2004), pp 411-425.
[16] Vallée, B. Dynamical Analysis of a Class of Euclidean Algorithms, Theoretical Computer Science, vol 297/1-3 (2003) pp 447–486.
[17] Vallée, B. Euclidean Dynamics, Discrete and Continuous Dynamical Systems, 15 (1) May 2006, pp 281-352.
[18] Vallée, B. Digits and Continuants in Euclidean Algorithms. Ergodic Versus Tauberian Theorems, Journal de Théorie des Nombres de Bordeaux 12 (2000) pp 531-570.

[19] Yap, C.K. *Fundamental Problems in Algorithmic Algebra*, Princeton University Press (1996).