

## Average Bit-Complexity of Euclidean Algorithms

*Brigitte Vallée*

GREYC, Université de Caen

May 22, 2000

*Summary by Marni Mishna*

### Abstract

The complexity the Euclidean algorithm and its variants is well studied. This work refines the problem further by considering precise average bit-complexity. The technique is sufficiently general as to apply to a wide class of gcd-type algorithms. It determines elementary transformations for each algorithm and derives asymptotic information from their analytic behaviour. The methods rely on properties of transfer operators adapted from dynamical systems theory. The use of Ergodic Theorems in the continuous case (continued fraction algorithms) foreshadows the results, which use Tauberian Theorems as replacement. This is joint work with Ali Akhavi [1].

### 1. Why the Bit Case?

Since the initial average case analysis of the Euclidean algorithm in 1969 by Heilbronn and Dixon a wide variety of approaches have been used to examine variants, the most recent of which is the method of using transfer operators [3, 4].

The technique involves viewing the algorithm as a dynamical system and each iterative step as a linear fractional transformation (LFT). Previous talks by the speaker [2] shed some light on this technique, how several classes of GCD algorithms fell under a unified approach and furthermore, why they were naturally divided into two categories: slow ( $\Theta(\log^2 n)$ ) and fast ( $\Theta(\log n)$ ).

This same technique will now aid in the study of bit-wise complexity. The motivation for this refinement is the following. It is not a priori evident whether the properties which make the slow algorithms slow extend to the bit case. It is true that there are more iterations, but what of the size of each iteration? This work answers the question definitively, yielding the same divisions between slow and fast algorithms, however with new complexity descriptions,  $\Theta(\log^3 n)$  and  $\Theta(\log^2 n)$ . Furthermore, it is of interest to consider a practical complexity measure. The method offers precise insights on the distribution of costs. This enables a further refinement on the classification between the fast and slow algorithms.

**1.1. Standard algorithm.** The standard Euclidean algorithm determines the gcd of  $v_0$  and  $v_1$  by a finite number of steps of the form  $v_i = m_i v_{i-1} + v_{i+1}$ , with final step  $v_k = 0$ . Define  $l(x) = \lfloor \log_2 x \rfloor + 1$ , the binary length of  $x$ . At each step there is one “naive” division with bit cost  $l(v_i)l(m_i)$ , and two assignment steps involving  $v_i$  and  $v_{i+1}$ . The total bit-complexity of one iteration is  $l(v_i)l(m_i) + l(v_i) + l(v_{i+1})$ . The cost for the standard algorithm is then

$$C(v_1, v_0) = \sum_{i=1}^k l(v_i) \cdot (l(m_i) + 2).$$

## 2. Main Result: Bit-Wise Complexity

The following two sets are valid input to the Euclidean algorithm:

$$\Omega = \{(u, v) \mid \gcd(u, v) = 1, 1 \leq u < v\} \quad \text{and} \quad \Omega_N = \{(u, v) \mid (u, v) \in \Omega, v \leq N\}.$$

The goal is to estimate the mean value of a cost  $C : \Omega \rightarrow \mathbb{R}$  on  $\Omega_N$ . More precisely, to determine the asymptotic value as  $N \rightarrow \infty$  of the mean value  $E_N[C]$  satisfying  $E_N[C] = C_N/|\Omega_N|$ , where  $C_N = \sum_{(u,v) \in \Omega_N} C(v, u)$ .

The function of interest in this presentation is the bit-cost of the standard Euclidean algorithm, and consequently the cost is as defined in the previous section, but the methods are sufficiently general as to apply to a number of cases. The technique views the algorithm as a dynamical system with each iterative step a LFT. Modifying the LFT yields the variants. The continued fraction expression of the problem motivates the use of the transformations  $U(x) = \frac{1}{x} - \lfloor \frac{1}{x} \rfloor$  and  $M(x) = \lfloor \frac{1}{x} \rfloor$ . Notice that  $m_{i+1} = M(U^i(v_1/v_0))$ . The value of  $k$  in the continued fraction to the right is the depth.

$$\frac{v_1}{v_0} = \frac{1}{m_1 + \frac{1}{m_2 + \frac{1}{m_3 + \cdots + \frac{1}{m_k}}}}$$

**2.1. Ergodic theory estimates.** Gauss observed that the iteration of the transformation  $U$  has invariant density  $\Psi(t) = \frac{1}{\log 2} \frac{1}{1+t}$ . For any  $A : \mathbb{N} \rightarrow \mathbb{R}$  such that  $\sum A(m)m^{-2} < \infty$ , define  $E_\infty[A(m)] = \int_0^1 A[m(t)]\Psi(t) dt$ . This is equal to

$$E_\infty[A(m)] = \sum_{m \geq 1} A(m) \left( \log_2 \left( 1 + \frac{1}{m} \right) - \log_2 \left( 1 + \frac{1}{m+1} \right) \right).$$

For example, when applied to  $l(m)$ :  $E_\infty[l(m)] = (1/\log 2) \log(\prod_{k \geq 1} 1 + 2^{-k})$ .

In the continuous case, ergodic theory is applicable and gives the result that the expected value  $E_N[\sum_{k=1}^m A(U^k(x))]$  approaches  $E_\infty[A]$  almost everywhere. Although ergodic theory does not apply in the discrete case, it does give plausible estimates as to what to expect. The assignment  $A(m) = l(m)$  gives the expected size of  $m_i$  in bits. The discrete version is formulated as  $E_N[\sum_{k=1}^{p(x)} A(U^k(x))]$ , where  $p(x)$  is the depth of the necessarily finite continued fraction expansion of the rational  $x$ . In this framework one can study the asymptotic behaviour of several functions on  $\Omega_N$ , such as:  $\tilde{A}(x) = \sum_{k=1}^{p(x)} A(m_k(x))$  and  $\tilde{C}(x) = \sum_{k=1}^{p(x)} l(m_k(x)) \cdot \log_2 v_k(x)$ .

One might anticipate that the value of  $E_N[\tilde{A}]$  under certain conditions should relate to the expected depth and the expected size of an iteration. The expected depth,  $\mathbf{E}[p]$ , corresponds to the number of iterations of the Euclidean algorithm on input  $\Omega_N$ , and is asymptotic to  $6/\pi^2 \log^2 N$ . So, in the case of  $A(m) = l(m)$ ,

$$E_N[\tilde{A}] \sim E_N[p] \times E_\infty[A(m)] = \left( \frac{12}{\pi^2} \log \prod_{k \geq 0} \left( 1 + \frac{1}{2^k} \right) \right) \log_2 N.$$

This is the mean size of the continued fraction encoding of a rational number. A similar heuristic analysis of  $E_N[\tilde{C}]$  shows the relation

$$E_N[\tilde{C}] \sim E_N[p] \frac{1}{2} \log_2 N \cdot E_\infty[l(m)].$$

These observations give a context for the main result.

**Theorem 1.** *The average bit-complexity of the standard Euclidean algorithm on the set of valid inputs of denominator less than  $N$  is asymptotically of log-squared order:*

$$E_N[C] \sim \left( \frac{6 \log 2}{\pi^2} \log \prod_{k \geq 1} \left( 1 + \frac{1}{2^k} \right) \right) \log_2^2 N.$$

This agrees with the heuristic argument. Numerically, this values satisfies  $E_N[C] \sim 1.24237 \log_2^2 N$ .

### 3. Summary of Methods

The general method for obtaining this result is similar to the speaker’s analysis of gcd-type functions. The average is expressible by partial sums of coefficients of Dirichlet series. Tauberian theory transfers the analytic behaviour of the series near singularities into asymptotic behaviour of coefficients. When seen as a dynamical system the generating functions of bit-cost relate to the Ruelle operators associated to the algorithm. The singularities of the Dirichlet series are related to spectral projections of the operators and are easy to describe.

**3.1. Dirichlet generating functions.** Define  $\omega_n$  to be the set of all pairs  $(u, v)$  in  $\Omega$  with  $v = n$  and  $C_n$  as the cumulative value of  $C$  over  $\omega_n$ . Then the corresponding encoding into Dirichlet generating functions is

$$F_{\langle c \rangle}(s) = \sum_{n \geq 1} \frac{C_n}{n^s} = \sum_{(v_0, v_1) \in \Omega} \frac{C(v_1, v_0)}{v_0^s}.$$

Thus the expected average cost is  $E_N[C] = (\sum_{n \leq N} C_n) / (\sum_{n \leq N} |\omega_n|)$ .

**3.2. Tauberian theorem.** The Tauberian theorems are a natural tool to consider as they give asymptotic information about the partial sums of coefficients of Dirichlet series. They rely on the nature and position of the singularities of  $F(s) = \sum a_n n^{-s}$ .

**Theorem 2** (Delange). *Let  $F(s)$  be a Dirichlet series with non-negative coefficients such that  $F(s)$  converges for  $\Re(s) > \sigma > 0$ . Assume that:*

1.  $F$  is analytic on  $\Re(s) = \sigma$ , where  $s \neq \sigma$ ;
2.  $F(s) = A(s)(s - \sigma)^{-\gamma-1} + C(s)$  for some  $\gamma \geq 0$ , and  $A(s)$  and  $C(s)$  analytic with  $A(\sigma) \neq 0$ .

Then, as  $N \rightarrow \infty$ , the partial sum of coefficients is

$$\sum_{n \leq N} a_n = \frac{A(\sigma)}{\sigma \Gamma(\gamma + 1)} N^\sigma \log^\gamma N (1 + \epsilon(N)), \quad \text{where } \epsilon(N) \rightarrow 0.$$

However, the conditions are difficult to verify for  $F_{\langle c \rangle}(s)$  in its present form. A transformation gives the required information about the singularities.

**3.3. Ruelle operators.** The classical operator is

$$G_s[F](x) = \sum_{m \geq 1} \frac{1}{(m+x)^s} F\left(\frac{1}{m+x}\right).$$

Let  $\mathcal{H} = \{h \mid h(x) = (m+x)^{-1}, m \geq 1\}$ , the set of inverse branches of  $U$ . If  $D[h]$  is the denominator of the LFT  $h(x)$ , then since  $D[h \circ g](x) = D[h]g(x) \cdot D[g](x)$ , the iterates of  $G_s$  are given by

$$G_s^k[F](x) = \sum_{h \in \mathcal{H}} \frac{1}{D[h](x)^s} F \circ h(x).$$

Rationals of  $\Omega$  can be written  $x = h(0)$  for some  $h$  in  $\mathcal{H}^k$  where  $k \geq 0$ . Then the Dirichlet generating function for  $|\omega_n|$  is equal to  $\sum_{n \geq 1} |\omega_n| n^{-s} = \sum_{h \in \mathcal{H}^*} D[h](0)^{-s} = (I - G_s)^{-1}[1](0)$ . A cost version of  $R_{s,h}[F](x) = D[h](x)^{-1} F \circ h(x)$  is defined as  $R_{s,h}^{[c]}[F](x) = c(h) D[h](x)^{-1} F \circ h(x)$ . Similarly the cost companion to  $G_s = \sum_{h \in \mathcal{H}} R_{s,h}$  is  $G_s^{[c]} = \sum_{h \in \mathcal{H}} R_{s,h}^{[c]}$ .

Recall that  $C(v_0, v_1) \sim \sum_{i=1} \log_2(v_i) c(m_i)$ . If  $x = v_1/v_0 = h_1 \circ h_2 \circ \dots \circ h_k(0)$ , then  $c(m_i)$  only depends on  $h_i$  and  $v_i$  only depends on  $h_{i+1} \circ \dots \circ h_k(0)$ . That is, the function can be expressed as

$$h = (h_1 \circ \dots \circ h_{i-1}) \circ h_i \circ (h_{i+1} \circ \dots \circ h_k) = b_i(h) \circ h_i \circ e_i(h).$$

$$\text{Defining } C_{s,h} = - \sum_{i=1}^k \frac{\partial}{\partial s} R_{s,e_i(h)} \circ R_{s,h_i}^{[c]} \circ R_{s,b_i(h)} \text{ yields } F_{\langle c \rangle}(s) = \sum_{h \in \mathcal{H}^*} C_{s,h}[1](0).$$

**3.4. Functional analysis.** The singularities of the cost function can now be described in terms of the singularities of the  $C_{s,h}$ , and subsequently of  $(I - G_s)^{-1}$ . Analysis of  $(I - G_s)^{-1}$  determines the values for the Tauberian theorem to be  $\sigma = 2$  and  $\gamma = 2$ . Using this, Theorem 1 now follows. In the case of the operators related to the slow algorithms, the corresponding result is  $\gamma = 3$ , accounting for the log-cubed behaviour.

#### 4. Variants and Encoding

As before, the technique applies to a family of variants. For example, the bit-complexity of the centred algorithm is asymptotic to

$$\frac{6 \log 2}{\pi^2} \log \left( \phi^2 \prod_{k=3}^{\infty} \frac{\phi^2 + \frac{2\phi}{2^k-1}}{\phi^2 - \frac{2}{2^k-1}} \right) \log_2^2 N, \quad \text{where } \phi = (\sqrt{5} + 1)/2.$$

Finally, the average length of a continued fraction encoding is computable. This is the room occupied in memory by  $(m_1, m_2, \dots, m_k, v_k)$ . The encoding uses the same principles as Fano–Shannon.

**Theorem 3.** *The average Fano–Shannon code-length  $D_N$  of the continued fraction expansion produced by the standard algorithm on valid inputs with denominator size  $N$  satisfies*

$$D_N \sim \frac{12 \log^2}{\pi^2} \left( 1 + \frac{2}{\log 2} \log \prod_{k=1}^{\infty} \left( 1 + \frac{1}{2^k} \right) \right) \log_2 N.$$

The numerical value is  $2.04868 \log_2 N$ , which is close to the optimal  $2 \log_2 N$ .

#### Bibliography

- [1] Akhavi (A.) and Vallée (B.). – Average bit-complexity of Euclidean algorithms. In Montanari (Ugo), Rolim (José D. P.), and Welzl (Emo) (editors), *Automata, languages and programming. Lecture Notes in Computer Science*, vol. 1853, pp. 374–387. – Springer, New York, 2000. Proceedings of the 27th ICALP Conference, Geneva, Switzerland, July 2000.
- [2] Salvy (B.). – *Algorithms Seminar, 1998–1999*. – Research Report n° 3830, Institut National de Recherche en Informatique et en Automatique, December 1999.
- [3] Vallée (B.). – Dynamics of the binary Euclidean algorithm: functional analysis and operators. *Algorithmica*, vol. 22, n° 4, 1998, pp. 660–685. – Average-case analysis of algorithms.
- [4] Vallée (Brigitte). – A unifying framework for the analysis of a class of Euclidean algorithms. In Gonnet (Gastón H.), Panario (Daniel), and Viola (Alfredo) (editors), *LATIN 2000: Theoretical Informatics. Lecture Notes in Computer Science*, vol. 1776, pp. 343–354. – Springer, Berlin, 2000. Proceedings of the 4th Latin American Symposium, Punta del Este, Uruguay, April 2000.