Gene Cooperman, Sandra Feisel, Joachim von zur Gathen & George Havas (1999). GCD of Many Integers. In *COCOON '99*, T. Asano, H. Imai, D. T. Lee, S. Nakano & T. Tokuyama, editors, number 1627 in Lecture Notes in Computer Science, 310–317. Springer-Verlag. ISSN 302-9743 (Print) 1611-3349 (Online). URL http://www.springerlink.com/content/gm5wjjcaaxx544w1/fulltext.pdf.
This document is provided as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that these works are posted here electronically. It is understood that all persons copying any of these documents will adhere to the terms and constraints invoked by each copyright holder, and in particular use them only for noncommercial purposes. These works may not be posted elsewhere without the explicit written permission of the copyright holder. (Last update 2016/05/18-14:15.)

submission ISSAC 98

GCD OF MANY INTEGERS

Gene Cooperman*, Sandra Feisel,
Joachim von zur Gathen and George Havas†

College of Computer Science
Northeastern University
Boston, MA 02115, USA
gene@ccs.neu.edu

FB Mathematik-Informatik, Universität–GH Paderborn
33095 Paderborn, Germany
{feisel,gathen}@uni-paderborn.de

School of Information Technology
The University of Queensland
Queensland 4072, Australia
havas@cs.uq.edu.au

Extended Abstract

Abstract. A probabilistic algorithm is exhibited that calculates the gcd of many integers using gcds of pairs of integers; the expected number of such gcds is less than two.

1. Introduction

In many algorithms for polynomials in $\mathbb{Z}[x]$, e.g., the computation of gcds or factorization, one is interested in making a polynomial primitive, i.e., computing and removing its content. The primitive polynomial remainder sequence needs such gcd computations but is not used in practice: Ho & Yap (1996) state that "the primitive PRS has the smallest possible coefficients, but it is not efficient because content computation is relatively expensive". Geddes

*Supported in part by NSF Grant CCR-9509783
†Supported in part by the Australian Research Council

et al. (1992), Chapter 7, write about the primitive polynomial remainder sequence: "The problem with this method, however, is that each step requires a significant number of GCD operations in the coefficient domain. [...] The extra cost is prohibitive when working over coefficient domains of multivariate polynomials."

We remedy this sorry situation for $\mathbb{Z}[x]$ by computing efficiently the gcd of random linear combinations of the inputs. More precisely, we solve the following problem.

> Given m positive integers $a_1, \ldots, a_m$ with $m > 1$, compute $\gcd(a_1, \ldots, a_m)$ with a small number of pairwise gcds, i.e., gcds of two integers each.

MAIN THEOREM. *This problem can be solved by taking an expected number of less than two pairwise* gcds *of random linear combinations of the input.*

The algorithm we state is quite natural. One would heuristically expect it to work quite well, and indeed our experimental results support the heuristic assumption. Unfortunately, we cannot *prove* what we expect to be true about it.

Our main contribution is a modification which makes the approach amenable to a precise analysis, and indeed we can then prove that its success probability comes arbitrarily close to what we conjecture to be true.

We stress that we are not doing an *average-case analysis* but rather look for a *provably reliable* method that works well no matter what the inputs are. To our knowledge, this problem has not been treated in the literature.

The use of random linear combinations is readily suggested by the success of that method for polynomials, but we do not know how to prove that the naive implementation of this idea works, although it does in practice. Our main algorithmic contribution is a clever choice of the range for the random coefficients. There is a natural (heuristic) upper bound on the success probability of any algorithm; with our choice of the range, we can prove a lower bound that is reasonably close to that upper bound. In fact, the lower bound can be moved arbitrarily close to the upper bound, but at the expense of requiring very large coefficients, which limits the practical usefulness.

However, we believe that our method, free of any heuristic or distributional assumption, makes the content computation for integer polynomials eminently practical.

2. Iterative gcd computation

By the associativity of the gcd

$$\begin{aligned}\gcd(a_1,\ldots,a_m) &= \gcd(\gcd(a_1,a_2),a_3,\ldots,a_m)\\ &= \gcd(\ldots(\gcd(\gcd(a_1,a_2),a_3),\ldots,a_m),\end{aligned}$$

the most obvious way to solve Problem 1 is to compute the pairwise gcds successively and to stop whenever a result is 1. This iterative computation of pairwise gcds will work well for random inputs, but there are "nasty" inputs on which that method will need $m-1$ pairwise gcd computations.

If the inputs are randomly chosen integers this naive algorithm does usually not need all the $m-1$ steps:

FACT 1. *Let two integers a and b be chosen uniformly at random from the positive integers up to N. Then for large N*

$$\text{prob}(\gcd(a,b)=1) \to \zeta(2)^{-1} = \frac{6}{\pi^2} \approx 0.6079271016.$$

For a proof, see Knuth (1981), Section 4.5.2.

In this case, the probability that a_1 and a_2 are already coprime is about 0.6, and for randomly chosen $a_1,\ldots,a_m$, we can therefore expect that the naive algorithm will already stop after about two computations of a pairwise gcd.

Heuristic reasoning says that a prime p divides a random integer a with probability p^{-1}, and m random integers with probability p^{-m}. Assuming the independence of these events, we have

$$\text{prob}(\gcd(a_1,\ldots,a_m)=1) = \prod_p (1-p^{-m}) = \zeta(m)^{-1}.$$

This can be made into a rigorous argument showing that the probability that m random integers, as in Fact 1, have trivial gcd tends (with $N\to\infty$) to $\zeta(m)^{-1}$. The first few values are:

m	2	3	4	5	6	7	8	9	10
$\zeta(m)^{-1}$	0.608	0.832	0.924	0.964	0.983	0.992	0.996	0.998	0.999

Although this strategy is quite successful for randomly chosen inputs, there exist "nasty" sequences, in which the $m-1$ steps in the above algorithm are really necessary.

EXAMPLE 2. *Let $p_1, \ldots, p_m$ be the first m primes, $A = p_1 \cdots p_m$, and $a_i = A/p_i$ for each i. Then $\gcd(a_1, \ldots, a_m) = 1$ but for any proper subset S of $\{1, \ldots, m\}$, $\gcd(\{a_i : i \in S\})$ is non-trivial. So the successive computation of pairwise gcds does not give the right output until $m - 1$ steps in the above algorithm have been performed.*

We do not address the question of representing the gcd as a linear combination of the inputs. This problem has been considered in Majewski & Havas (1995).

3. Random linear combinations

In the case of polynomials over a field F, the use of random linear combinations is known to be successful.

FACT 3. *Let F be a field, $a_1, \ldots, a_m \in F[x]$ be nonzero polynomials of degree at most d, $h = \gcd(a_1, \ldots, a_m)$, $A \subseteq F$ finite, $x_3, \ldots, x_m \in A$ be randomly chosen elements, and $g = a_2 + \sum_{3 \le i \le m} x_i a_i \in F[x]$. Then h divides $\gcd(a_1, g)$, and*

$$\text{prob}(h = \gcd(a_1, g)) \ge 1 - d/\#A.$$

This is based, of course, on resultant theory (Díaz & Kaltofen (1995), von zur Gathen *et al.* (1996)), and also works for multivariate polynomials. So for the polynomial case, we expect that with only about one calculation of a pairwise gcd of two polynomials, we find the true gcd of many polynomials provided that A is chosen large enough.

The following algorithm is a natural adaptation to integers.

ALGORITHM 4. *Probabilistic gcd of many integers.*
Input: *m positive integers $a_1, \ldots, a_m$ with $a_i \le N$ for all i, and a positive integer M.*
Output: *Probably* $\gcd(a_1, \ldots, a_m)$.

1. *Pick $2m$ uniformly distributed random integers $x_1, \ldots, x_m$ and $y_1, \ldots, y_m$ in $\{1, \ldots, M\}$, and compute $x = \sum_{1 \le i \le m} x_i a_i$ and $y = \sum_{1 \le i \le m} y_i a_i$.*
2. *Return* $\gcd(x, y)$.

Then $g = \gcd(a_1, \ldots, a_m)$ divides $\gcd(x, y)$, and we can easily check equality by trial divisions by $\gcd(x, y)$. This is a Monte-Carlo algorithm and we want to prove a lower bound for its success probability, i.e., for the probability that $\gcd(x, y)$ equals g.

Due to Fact 1 we cannot expect this probability to be as high as in the polynomial case, but experiments show that this strategy seems to have roughly the expected success probability of $6/\pi^2$. For two lists we have tested whether two random linear combinations of the list elements have the same gcd as the list elements. The list "nasty" has 100 elements as described in Example 2 and whose largest entry has 220 decimal digits, and "rand" is a random lists with 100 elements of up to 220 decimal digits. Each success rate is the average of 5000 independent tests.

Table 1: random linear combinations for two input sequences.

list	M	success rate	list	M	success rate
nasty	2	0.7542	rand	2	0.6066
nasty	10	0.6522	rand	10	0.6016
nasty	100	0.6146	rand	100	0.6078
nasty	1000	0.602	rand	1000	0.6098
nasty	30030	0.5952	rand	30030	0.6038

Table 1 shows that one gcd computation of two random linear combinations of the a_i gives the right answer in about 60% of the cases, i.e., in most of the cases, and this seems to be somewhat independent of the size of M.

Our question is: how we can *prove* the probability that $\gcd(x, y) = g$ to be high.

The following experiment shows that some prudence is indicated in these matters. We adapted the algorithm from Fact 3 literally to integers and ran it on the "nasty" input, sorted by size. Then the success probability dropped dramatically to only 5%; on the "rand" list, it sagged to 30%.

Remembering Fact 1, the solution of the problem would be easy if x and y were random numbers, but although the results in Table 1 show that they behave approximatively as if they were uniformly distributed, this is not literally true. So we have to find another way to bound the probability that $g \neq \gcd(x, y)$.

4. A probabilistic estimate

It would, of course, be sufficient to show that our random linear combination x behaves like a random integer in its range. Then the bound $\zeta(s)^{-1}$ would hold for the gcd of s random linear combinations. In any case, we cannot reasonably hope to have a better success probability than this bound. However, we want a (small) bound on M, possibly in terms of the inputs. Then, if $a_1, \ldots, a_m \leq N$, we have

$$x = \sum_{1 \leq i \leq m} x_i a_i \leq mMN = B,$$

but x is clearly not uniformly distributed in $\{1, \ldots, B\}$. Even if it were, the probability that a prime p divides x would not be the exactly desired $1/p$, but only close to it, since B is not necessarily a multiple of p. We circumvent this obstacle by choosing M to be the product of the first r primes. For $r = 6$, we have $M = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 = 30030$. (In fact, for our purposes also a multiple of this number suffices.) Then for all primes p up to the rth prime p_r, p divides a random linear combination with probability exactly $1/p$. For our probability estimate, the main contribution becomes

$$\prod_{p \leq p_r} (1 - p^{-s}),$$

just as for the unavoidable $\zeta(s)^{-1} = \prod_p (1 - p^{-s})$. We only execute this idea for $s = 2$, and abbreviate

$$\eta_r = \prod_{p \leq p_r} (1 - p^{-2}).$$

We fix the following notation: Let $m \geq 2, r \geq 1$, let $a_1, \ldots, a_m$ be positive integers, all at most N, M a multiple of $p_1 \cdots p_r$, and let $x_1, \ldots, x_m, y_1, \ldots, y_m$ be uniformly distributed random integers between 1 and M, $x = \sum_{1 \leq i \leq m} x_i a_i$ and $y = \sum_{1 \leq i \leq m} y_i a_i$.

For the probability estimate we need the following lemma:

LEMMA 5. *If* $\gcd(a_1, \ldots, a_m) = 1$, *then the events that* p_k *divides* x *for* $1 \leq k \leq r$ *are independent.*

PROOF. Let $1 \leq k \leq r$. Since $\gcd(a_1, \ldots, a_m) = 1$, there exists an index j such that $p_k \nmid a_j$. Then for any $x_1, \ldots x_{j-1}, x_{j+1}, \ldots, x_m \in \mathbb{Z}$ the congruence

$$x_j a_j \equiv -\sum_{i \neq j} x_i a_i \bmod p_k$$

has exactly one solution x_j modulo p_k. Hence, there are p_k^{m-1} solutions with $1 \le x_1, \ldots, x_m \le p_k$ and

$$\sum_{1 \le i \le m} x_i a_i \equiv 0 \bmod p_k.$$

Now let $I \subseteq \{1, \ldots, r\}$ and $q = \prod_{k \in I} p_k$. By the Chinese Remainder Theorem, we have $\prod_{k \in I} p_k^{m-1} = q^{m-1}$ solutions modulo q giving $x \equiv 0 \bmod q$. Since $q \mid M$, this congruence has $q^{m-1} \cdot (M/q)^m$ solutions $1 \le x_1, \ldots, x_m \le M$. Hence,

$$\operatorname{prob}\{q \mid x\} = \frac{q^{m-1} \cdot \left(\frac{M}{q}\right)^m}{M^m} = \frac{1}{q}. \tag{1}$$

In particular, $\operatorname{prob}\{p_k \mid x\} = 1/p_k$ for each $k \le r$, and since (1) holds for each subset of $\{1, \ldots, r\}$, the events are independent. □

THEOREM 6. *With the above notation, let P be the probability that* $\gcd(a_1, \ldots, a_m) = \gcd(x, y)$. *Then*

$$P > \eta_r - \frac{2.04}{p_{r+1} \ln p_{r+1}} - \frac{2}{M}\left(\ln\ln M + \frac{1}{\ln^2 M} + \frac{1}{2\ln^2 p_r} - \ln\ln p_r\right) + \frac{r}{M^2} - \frac{mN}{M(\ln(mMN) - 3/2)}.$$

PROOF. We first assume that $\gcd(a_1, \ldots, a_m) = 1$. In the following, p is a parameter ranging over the prime numbers. Let α_p be the probability that p divides x. As in the proof of the lemma, there is an index $j \le m$ with $p \nmid a_j$. If we fix j, we find that the congruence

$$x_j a_j \equiv -\sum_{i \ne j} x_i a_i \bmod p$$

has at most one solution for x_j modulo p if the x_i with $i \ne j$ have already been chosen. There are M^{m-1} choices for those x_i. Since there is one possibility for x_j modulo p to solve this congruence, there are at most $\lceil \frac{M}{p} \rceil$ choices for x_j between 1 and M. Hence,

$$\alpha_p \le \frac{M^{m-1} \cdot \lceil \frac{M}{p} \rceil}{M^m} = \frac{\lceil \frac{M}{p} \rceil}{M}.$$

This bound is close but not equal to p^{-1}. Obviously, α_p is also the probability for p to divide y, and since the x_i and y_i are chosen independently, p divides

both x and y with probability α_p^2. Then the probability P that there is no prime which divides x and y is at least $1 - \sum_{p \le B} \alpha_p^2$, where $B = mNM$. It suffices to regard primes up to B since this is an upper bound for x and y.

We distinguish three cases for such primes p. All the estimates are valid for any value of M except that our special choice of M plays a role in case 1, but that is the case that provides the dominant term.

Case 1: $p \le p_r$. Then p divides M and $\alpha_p \le 1/p$ (in fact, equality holds).

Case 2: $p_r < p \le M$. Then $\alpha_p \le \frac{(M/p)+1}{M} = 1/p + 1/M$.

Case 3: $p > M$. Then $\alpha_p \le \frac{1}{M}$.

Furthermore, we know by Lemma 5 that the events that p divides x are independent for $p \le p_r$, i.e., for these small p's the probabilities are multiplicative. We obtain

$$\begin{aligned} 1 - P &\le \text{prob}(\exists p \le p_r : p \mid x \text{ and } p \mid y) + \text{prob}(\exists p > p_r : p \mid x \text{ and } p \mid y) \\ &\le 1 - \text{prob}(\forall p \le p_r : p \nmid x \text{ or } p \nmid y) + \sum_{p > p_r} \alpha_p^2 \\ &= 1 - \prod_{1 \le i \le r} (1 - \frac{1}{p_i^2}) + \sum_{p > p_r} \alpha_p^2 \\ &= 1 - \eta_r + \sum_{p > p_r} \alpha_p^2. \end{aligned}$$

We now use several bounds from Rosser & Schoenfeld (1962): Inequalities (3.17) and (3.20) for $\sum_{p_r < p \le M} 1/p$, Theorem 2 for the number $\pi(B)$ of primes up to B, and an estimate on p. 87 for $\sum_{p_r < p} 1/p^2$.

$$\begin{aligned} \sum_{p > p_r} \alpha_p^2 &\le \sum_{p_r < p \le M} \frac{1}{p^2} + \frac{2}{M} \sum_{p_r < p \le M} \frac{1}{p} + \sum_{p_r < p \le B} \frac{1}{M^2} \\ &< \sum_{p_r < p} \frac{1}{p^2} + \frac{2}{M} \left(\ln\ln M + \frac{1}{\ln^2 M} + \frac{1}{2\ln^2 p_r} - \ln\ln p_r \right) + \frac{\pi(B) - r}{M^2} \\ &< \frac{2.04}{p_{r+1} \ln p_{r+1}} + \frac{2}{M} \left(\ln\ln M + \frac{1}{\ln^2 M} + \frac{1}{2\ln^2 p_r} - \ln\ln p_r \right) \\ &\quad - \frac{r}{M^2} + \frac{B}{M^2(\ln(B) - 3/2)}. \end{aligned}$$

Substituting $B = mMN$ we obtain

$$\begin{aligned} P &> \eta_r - \frac{2.04}{p_{r+1}\ln p_{r+1}} - \frac{2}{M}\left(\ln\ln M + \frac{1}{\ln^2 M} + \frac{1}{2\ln^2 p_r} - \ln\ln p_r\right) \\ &+ \frac{r}{M^2} - \frac{mN}{M(\ln(mMN) - 3/2)}, \end{aligned}$$

as claimed. If $g = \gcd(a_1, \ldots, a_m) \neq 1$, the bound holds for $a_1/g, \ldots, a_m/g$ with N replaced by N/g. □

We note that only the last summand in the lower bound is input driven; it is easy to choose M so that it becomes arbitrarily small. The other terms only depend on our choices of r and M in the algorithm. Since η_r tends to $\zeta(2)^{-1}$, the lower bound can be brought arbitrarily close to that limit.

COROLLARY 7. *For any $\epsilon > 0$ one can choose r and M such that $P > \zeta(2)^{-1} - \epsilon$.*

This is a satisfying result, since we have no right to expect a better bound than the $\zeta(2)^{-1}$, which holds (in the limit) for truly random values.

COROLLARY 8. *Let $r \geq 6$ and M be a multiple of $p_1 \cdots p_r$. Then*

$$P > 0.5756 - \frac{mN}{M(\ln(mMN) - 3/2)}.$$

PROOF. We have $\eta_r > \zeta(2)^{-1}$ for all $r \geq 1$, and

$$\frac{2}{M}\left(\ln\ln M + \frac{1}{\ln^2 M} + \frac{1}{2\ln^2 p_r} - \ln\ln p_r\right) < 0.0000983402$$

for $r = 6$ and $M = p_1 \cdots p_6 = 30030$. The left hand side decreases strictly for increasing r. Hence,

$$\begin{aligned} P &> 0.6079271016 - \frac{2.04}{17\ln 17} - 0.0000983402 - \frac{mN}{M(\ln(mMN) - 3/2)} \\ &> 0.5756253577 - \frac{mN}{M(\ln(mMN) - 3/2)}. \quad \square \end{aligned}$$

COROLLARY 9. *Let $r \geq 6$, and $M \geq 10000mN/(51 + 6\ln(mN))$ be a multiple of $p_1 \cdots p_r$. Then $P > 57.5\%$.*

PROOF. We have $\ln(M) \geq \ln(30030) > 10$ and

$$\begin{aligned} P &> 0.5756 - \frac{mN(51 + 6\ln(mN))}{10000mN(\ln(M) + \ln(mN) - 3/2)} \\ &> 0.5756 - \frac{6(8.5 + \ln(mN))}{10000(8.5 + \ln(mN))} \\ &= 0.5756 - \frac{6}{10000} = 0.575. \quad \square \end{aligned}$$

So, if M is chosen large enough, Algorithm 4 has a success probability of about 0.57. The M chosen in Table 1 is far too small compared to the M suggested in Corollary 9. The results in that table suggest that P is even somewhat larger than in the corollary and rather independent both of the size of M and of its choice as a multiple of the small primes, but we do not know how to prove anything about this.

Acknowledgements

We are grateful to Mark Giesbrecht and Boaz Patt-Shamir for helpful discussions.

References

ANGEL DÍAZ AND ERICH KALTOFEN, On computing greatest common divisors with polynomials given by black boxes for their evaluations. In *Proc. Int. Symp. Symbolic and Algebraic Computation ISSAC '95, Montréal, Canada*, 1995, 232–239.

J. VON ZUR GATHEN, M. KARPINSKI, AND I. E. SHPARLINSKI, Counting curves and their projections. *Computational complexity* **6** (1996), 64–99. Extended Abstract in Proc. 25th ACM Symp. Theory of Computing.

K. O. GEDDES, S. R. CZAPOR, AND G. LABAHN, *Algorithms for Computer Algebra*. Kluwer Academic Publishers, 1992.

CHUNG-YEN HO AND CHEE KENG YAP, The Habicht approach to subresultants. *Journal of Symbolic Computation* **21** (1996), 1–14.

DONALD E. KNUTH, *The Art of Computer Programming, Vol.2, Seminumerical Algorithms*. Addison-Wesley, Reading MA, 2 edition, 1981.

BOHDAN S. MAJEWSKI AND GEORGE HAVAS, A solution to the extended gcd problem. In *Proc. ISSAC '95*. ACM Press, 1995, 248–253.

J. B. Rosser and L. Schoenfeld, Approximate formulas for some functions of prime numbers. *Ill. J. Math.* **6** (1962), 64–94.

R. Zippel, *Effective polynomial computation*. Kluwer Academic Publishers, 1993.