A New Algorithm and Refined Bounds for Extended Gcd Computation

David Ford* and George Havas**

Department of Computer Science,
Concordia University, Montréal, Québec, Canada H3G 1M8
and
Department of Computer Science,
The University of Queensland, Queensland 4072, Australia

Abstract. Extended gcd computation is interesting itself. It also plays a fundamental role in other calculations. We present a new algorithm for solving the extended gcd problem. This algorithm has a particularly simple description and is practical. It also provides refined bounds on the size of the multipliers obtained.

1 Introduction

The gcd problem for more than two numbers is interesting in its own right (see, for example, the research level problem in [8]). Furthermore, it has important applications in its extended form where corresponding multipliers are required, for example in computing canonical normal forms of integer matrices ([3, 4, 7]).

Some earlier algorithms for computing the greatest common divisor of more than two numbers include those of [1, 2, 7, 14]. More recent algorithms and analyses provide a much better understanding of the area. On the one hand we have complexity results which show that there are a number of problems for which efficient solutions are not always readily available [5, 9]. These include: given a multiset of numbers, how many of them do we need to take to obtain the gcd of all of them? how can we efficiently find 'good' multipliers in an extended gcd computation? how quickly can we obtain the result? On the other hand we have a selection of theoretical [9] and practical algorithms [5, 6, 10, 11] which have been analysed and used in various applications.

We present a new algorithm with a particularly simple description. The algorithm is readily implemented. It also provides refined bounds on the size of the multipliers obtained.

The problem that we consider is to express the gcd of n integers, for arbitrary n, as an integer linear combination of the numbers. It is convenient to order the multiset of input integers so that we can use linear algebra to describe the

* Email: ford@cs.concordia.ca; partially supported by the Natural Sciences and Engineering Research Council (Canada) and Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (Québec).

** Email: havas@cs.uq.edu.au; partially supported by the Australian Research Council.

problem and solution. Thus, given a column vector A we wish to find a row vector X such that $XA = \gcd_i(a_i)$. The difficulty arises in finding "good" vectors X. In particular we have the following result [9]: finding an optimal solution X with respect to either the L_0 metric or the L_∞ norm is NP-hard. (The L_0 metric measures sparsity while the L_∞ norm is the max norm.)

2 The New Algorithm

The new algorithm arises from a study of the gcd-tree method in [9]. Basically we construct a matrix P which has a solution to the extended gcd problem for A as its first row and a basis for the null space of A as its remaining rows. We then row reduce the first row of P (using a least remainder reduction) to give an improved solution.

Define $g_k = \gcd(a_k, \ldots, a_n)$, for $1 \leq k \leq n$, and let

$$A = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{n-1} \\ a_n \end{pmatrix}, \quad B = \begin{pmatrix} g_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} a_1 & 0 & 0 & \cdots & 0 & 0 \\ a_2 & a_2 & 0 & \cdots & 0 & 0 \\ a_3 & a_3 & a_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_{n-1} & a_{n-1} & a_{n-1} & \cdots & a_{n-1} & 0 \\ a_n & a_n & a_n & \cdots & a_n & a_n \end{pmatrix}.$$

Consider the $n \times n$ integer matrix

$$P = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & \cdots & c_{1,n-1} & c_{1,n} \\ -\frac{g_2}{g_1} & \frac{a_1 c_{2,2}}{g_1} & \frac{a_1 c_{2,3}}{g_1} & \cdots & \frac{a_1 c_{2,n-1}}{g_1} & \frac{a_1 c_{2,n}}{g_1} \\ 0 & -\frac{g_3}{g_2} & \frac{a_2 c_{3,3}}{g_2} & \cdots & \frac{a_2 c_{3,n-1}}{g_2} & \frac{a_2 c_{3,n}}{g_2} \\ 0 & 0 & -\frac{g_4}{g_3} & \cdots & \frac{a_3 c_{4,n-1}}{g_3} & \frac{a_3 c_{4,n}}{g_3} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{g_n}{g_{n-1}} & \frac{a_{n-1} c_{n,n}}{g_{n-1}} \end{pmatrix}$$

where $\sum_{j=k}^{n} c_{k,j} a_j = g_k$.

Then $PA = B$, and because

$$PC = \begin{pmatrix} g_1 & * & * & \cdots & * & * \\ 0 & \frac{a_1 g_2}{g_1} & * & \cdots & * & * \\ 0 & 0 & \frac{a_2 g_3}{g_2} & \cdots & * & * \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{a_{n-2} g_{n-1}}{g_{n-2}} & * \\ 0 & 0 & 0 & \cdots & 0 & \frac{a_{n-1} g_n}{g_{n-1}} \end{pmatrix}.$$

we have $\det(PC) = \det(C)$, so that $\det(P) = +1$.

The important thing to notice about this construction is that any matrix P satisfying the requirements suffices as a starting point. One such matrix is readily obtained by using a straightforward recursive extended gcd algorithm. Thus, in this approach, the first step is to compute the 2×2 matrix comprising the rightmost entries of the top and bottom rows. Next, extend this to the 3×3 matrix comprising the rightmost entries of the top and the bottom two rows, and so on.

3 Analysis

A general upper bound on the size of the multipliers (with respect to the max norm) is given by a theorem in [9].

Theorem 1. *There is an optimal time, optimal space algorithm for computing the extended gcd of n integers $\{a_1, \ldots, a_n\}$, which guarantees that no multiplier is larger than the largest of the numbers divided by 2.*

We refine that bound via an analysis of this new algorithm to obtain the following result.

Theorem 2. *Let $\{a_1, \ldots, a_n\}$ be a multiset of positive integers. Let $m = \max(a_1, \ldots, a_{n-1})$ and let $g_k = \gcd(a_k, \ldots, a_n)$ for $1 \leq k \leq n$. Then there exists an integer solution to the equation*

$$x_1 a_1 + \cdots + x_n a_n = g_1$$

which satisfies

$$-\frac{g_{j+1}}{2g_j} < x_j \leq \frac{g_{j+1}}{2g_j} \text{ for } 1 \leq j \leq n-1, \tag{1}$$

$$|x_n| \leq \max\left(\frac{m}{2g_1}, 1\right). \tag{2}$$

Proof: Part (1) is established by row-reducing the first row of the matrix P, then setting $x_1 = P_{1,1}, \ldots, x_n = P_{1,n}$. The proof of (2) has three cases.

1. $g_n = g_1$.
   Then $x_1 = \cdots = x_{n-1} = 0$; hence $x_n = g_1/a_n = 1$.
2. $g_n = 2g_1$.
   Then for some j we have $x_j a_j + x_n a_n = g_1$, with $g_{j+1} = 2g_j$ and $g_{i+1} = g_i$ when $i \neq j$. It follows that $x_j = 1$, which gives $|x_n| = (a_j - g_1)/(2g_1) < m/(2g_1)$.
3. $g_n > 2g_1$.
   We have $x_j = 0$ whenever $g_j = g_{j+1}$; consequently

$$\sum_{j=1}^{n-1} |x_j| = \sum_{g_j < g_{j+1}} |x_j| \leq \frac{1}{2} \sum_{g_j < g_{j+1}} \frac{g_{j+1}}{g_j} \leq \frac{1}{2} \prod_{g_j < g_{j+1}} \frac{g_{j+1}}{g_j} = \frac{g_n}{2g_1}.$$

   It follows that

$$\begin{aligned} |x_n|\, g_n = |x_n a_n| &= |\, g_1 - x_1 a_1 - \cdots - x_{n-1} a_{n-1}\,| \\ &\leq g_1 + a_1|x_1| + \cdots + a_{n-1}|x_{n-1}| \\ &\leq g_1 + m\big(|x_1| + \cdots + |x_{n-1}|\big) \\ &\leq g_1 + m\left(\frac{g_n}{2g_1}\right); \\ 2|x_n| &\leq \frac{2g_1}{g_n} + \frac{m}{g_1}. \end{aligned}$$

   Now $2|x_n|$ and m/g_1 are integers, and $0 < 2g_1/g_n < 1$. Therefore $2|x_n| \leq m/g_1$, and hence $|x_n| \leq m/(2g_1)$. □

Note that the bounds given here for the multipliers are optimal in the sense that any individual multiplier can be made to approach its bound as closely as desired. This can be seen by considering the example with repeated numbers provided by the multiset $\{2, 2k+1, 2k+1, \ldots, 2k+1\}$, with $k \geq 1$, in various orders. We also point out that this bound is not worse than about the square of the best possible bound for distinct numbers, since it is shown in [6] that a general lower bound for the Euclidean norm of the multiplier vector in terms of the initial numbers a_i must be at least $O(\sqrt{\max\{a_i\}})$.

Furthermore we also provide bounds for the size of entries in the basis for the associated null space.

Corollary 3. *There is a unimodular transforming matrix P satisfying $PA = B$ with entries $p_{k,j}$ which satisfy*

$$-\frac{g_{j+1}}{2g_j} < p_{k,j} \leq \frac{g_{j+1}}{2g_j} \quad \text{for } 1 \leq k \leq j \leq n-1,$$

$$|p_{k,n}| \leq \max\left(\frac{a_k}{2g_k}, \ldots, \frac{a_{n-1}}{2g_k}, 1\right) \quad \text{for } 1 \leq k \leq n.$$

This result is useful for situations where the transforming matrix is required.

4 Concluding Remarks

We have presented a new algorithm for solving the extended gcd problem which is simply described and practical. It yields good bounds on the size of the multipliers obtained.

The algorithm can be directly used in the computation of canonical forms of matrices. The first author has investigated the computation of the rational canonical form. It must be noted that any "exact" method that proceeds by successive substitutions of the form $A \leftarrow T^{-1}AT$, with T a unimodular matrix producing the gcd g of elements $a_1, \ldots, a_n$ in a given row of A, must necessarily be of complexity exponential in n. The problem is that the values $a_1/g, \ldots, a_n/g$ must appear in T^{-1}, and so the maximum size of the entries of A can double at each step. The p-adic method of [12] avoids this difficulty, as does the exact method of Ozello [13]. The Ozello algorithm relies on the computation of dependence relations among integer vectors, and hence might be subject to improvement by application of our extended gcd algorithm. This possibility remains to be investigated.

References

1. W.A. Blankinship. A new version of the Euclidean algorithm. *Amer. Math. Mon.*, 70:742–745, 1963.
2. G.H. Bradley. Algorithm and bound for the greatest common divisor of n integers. *Commun. ACM*, 13:433–436, 1970.
3. G. Havas and B.S. Majewski. Integer matrix diagonalization. *J. Symbolic Comput.*, to appear.
4. G. Havas and B.S. Majewski. Hermite normal form computation for integer matrices. *Congressus Numerantium*, 105:184–193, 1994.
5. G. Havas and B.S. Majewski. A hard problem which is almost always easy. In *Algorithms and Computation*, Lecture Notes in Computer Science 1004, 216–223, 1995.
6. G. Havas, B.S. Majewski and K.R. Matthews. Extended gcd algorithms. Technical Report TR0302, The University of Queensland, Brisbane, 1994.
7. C.S. Iliopoulos. Worst case complexity bounds on algorithms for computing the canonical structure of finite abelian groups and the Hermite and Smith normal forms of an integer matrix. *SIAM J. Computing*, 18:658–669, 1989.
8. D.E. Knuth. *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms.* Addison-Wesley, Reading, Mass., 2nd edition, 1973.
9. B.S. Majewski and G. Havas. The complexity of greatest common divisor computations. In *Algorithmic Number Theory*, Lecture Notes in Computer Science 877, 184–193, 1994.
10. B.S. Majewski and G. Havas. A solution to the extended gcd problem. In *ISSAC'95 (Proc. 1995 Internat. Sympos. Symbolic Algebraic Comput.)*, ACM Press, New York, 248–253, 1995.
11. B.S. Majewski and G. Havas. Extended gcd calculation. *Congressus Numerantium*, 111:104–114, 1995.
12. M-H. Mathieu and D. Ford. On p-adic Computation of the Rational Form of a Matrix, *J. Symbolic Comput.*, 10:453-464, 1990.

13. P. Ozello. *Calcul Exact des formes de Jordan et de Frobenius d'une Matrice.* Doctoral Thesis, University of Grenoble, 1987.
14. M.S. Waterman. Multidimensional greatest common divisor and Lehmer algorithms. *BIT*, 17:465–478, 1977.