All the sequences above are complete (although repetitions must be permitted in Example 1 if $a > 2$), but the theorem does not assume completeness. We conclude with an example of a sequence which is not complete but by an immediate application of Corollary 1 is seen to be transformed into $\mathbb{N}$ under $f(x) = \ln x$. The sequence in question is $s_1 = 1$, $s_2 = 2$, and for $n \geq 3$, $s_n = 5 \cdot 2^{n-3}$. To see that this sequence is not complete, observe that

$$5 \cdot 2^{n-3} - 1, \text{ for } n \geq 3,$$

can never be expressed as the sum of distinct terms of the sequence.

Finally, we would like to sincerely thank Professor Gerald E. Bergum for suggesting many improvements in the content and presentation of this article.

REFERENCES

1. H. L. Alder. "The Number System in More General Scales." *Mathematics Magazine* 35 (1962):145-151.
2. John L. Brown, Jr. "Note on Complete Sequences of Integers." *The American Math. Monthly* 68 (1961):557-560.
3. John L. Brown, Jr. "Some Sequence-to-Sequence Transformations Which Preserve Completeness." *The Fibonacci Quarterly* 16 (1978):19-22.
4. V. E. Hoggatt, Jr. & C. H. King. "Problem E-1424." *The American Math. Monthly* 67 (1960):593.
5. Waclaw Sierpinski. *Elementary Theory of Numbers*. Warszawa, 1964.

*****

FINDING THE GENERAL SOLUTION OF A LINEAR DIOPHANTINE EQUATION

SUSUMU MORITO and HARVEY M. SALKIN*
Case Western Reserve University, Cleveland, OH 44106

ABSTRACT

A new procedure for finding the general solution of a linear diophantine equation is given. As a byproduct, the algorithm finds the greatest common divisor (gcd) of a set of integers. Related results and discussion concerning existing procedures are also given.

1. INTRODUCTION

This note presents an alternative procedure for computing the greatest common divisor of a set of n integers $a_1, a_2, \ldots, a_n$, denoted by

$$\gcd(a_1, a_2, \ldots, a_n),$$

---

*The authors would like to express their appreciation to Professor Dong Hoon Lee (Department of Mathematics, Case Western Reserve University) for his time and helpful discussions.

Part of this work was supported by the Office of Naval Research under contract number N00014-67-A-0404-0010.

and for finding the general solution of a linear diophantine equation in which these integers appear as coefficients. A classical procedure for finding the gcd of integers is based on the repeated application of the standard Euclidean Algorithm for finding the gcd of two integers. More specifically, it repeatedly uses the argument:

$$\gcd(a_1, a_2, \ldots, a_n) = \gcd(\gcd(a_1, a_2), a_3, \ldots, a_n).$$

A more efficient algorithm, which is related to the procedure presented here for computing the gcd was given by Blankinship [1]. Weinstock [2] developed a procedure for finding a solution of a linear diophantine equation, and Bond [3] later showed that the Weinstock Algorithm can be applied repeatedly to find the general solution of a linear diophantine equation.

In this note, we present an alternative approach to finding the general solution, and show that the algorithm produces $(n - 1)$ n-dimensional vectors with integer components whose integer linear combination generates all solutions which satisfy the linear diophantine equation with the right-hand side 0. We call a set of these $(n - 1)$ "generating" vectors a generator. It is easy to show that the generator is not unique for $n \geq 3$. In fact, for $n \geq 3$ there exist infinitely many generators. The proposed algorithm has certain desirable characteristics for computer implementation compared to the Bond Algorithm. Specifically, the Bond Algorithm generally produces generating vectors whose (integer) components are mostly huge numbers (in absolute values). This often makes computer implementation unwieldy [5]. The approach, presented here, was initially suggested by Walter Chase of the Naval Ocean Systems Center, San Diego, California, in a slightly different form for solving the radio frequency intermodulation problem [4].

For illustrative purposes, we will continuously use the following example with $n = 3$:

$$(a_1, a_2, a_3) = (8913, 5677, 4378).$$

Or, we are interested in the generator of:

$$8913x_1 + 5677x_2 + 4378x_3 = 0.$$

It turns out that the Bond Algorithm [3] produces the two generating vectors (5677, -8913, 0) and (2219646, 3484888, -1), whereas the procedure we propose gives (cf. Section 3) (-57, 17, 94) and (61, -95, -1).

Three obvious results are given without proof. Throughout this paper, we assume that the right-hand side of a linear diophantine equation a_0, if it is nonzero, is an integer multiple of $d = \gcd(a_1, a_2, \ldots, a_n)$. This is because of the well-known result [6] which says that a linear diophantine equation has a solution if and only if a_0 is divisible by d, and if d divides a_0 there are an infinite number of solutions.

Lemma 1: Consider the following two equations:

(1) $$a_1x_1 + a_2x_2 + \cdots + a_nx_n = 0;$$

(2) $$a_1x_1 + a_2x_2 + \cdots + a_nx_n = a_0.$$

Assume that $(x_{F_1}, \ldots, x_{F_{n-1}})$ is the generator of (1). Then, all solutions $x = (x_i)$ of (2) can be expressed in the form

(3) $$x = x^0 + k_1x_{F_1} + k_2x_{F_2} + \cdots + k_{n-1}x_{F_{n-1}},$$

where x^0 is any solution satisfying (2) and $k_1, k_2, \ldots, k_{n-1}$ are any integers.

Lemma 2: If $a_1' = a_1 + \ell_2 a_2 + \ell_3 a_3 + \cdots + \ell_n a_n$ for some integers $\ell_2, \ell_3, \ldots, \ell_n$, then gcd $(a_1, a_2, \ldots, a_n)$ = gcd $(a_1', a_2, \ldots, a_n)$.

Lemma 3: If $a_1 + \ell_2 a_2 + \ell_3 a_3 + \cdots + \ell_n a_n = 0$ for some integers $\ell_2, \ell_3, \ldots, \ell_n$, then gcd $(a_1, a_2, \ldots, a_n)$ = gcd $(a_2, \ldots, a_n)$.

Notice, for example, Lemma 3 is true because if

$$d = \gcd(a_2, \ldots, a_n)$$

then

$$a_1 = \left(\sum_{i=2}^{n} \ell_i'\right) \cdot d, \text{ for some integers } \ell_i' (2 \le i \le n).$$

Thus,

$$\gcd(a_1, a_2, \ldots, a_n) = \gcd(a_1, d) = d.$$

Finding the general solution of a linear diophantine equation having a right-hand side different from zero (say $a_0 \neq 0$) is straightforward, because of Lemma 1, if the generator and one solution for (2) is known. The algorithm we propose first finds a solution, say x^d, for the linear diophantine equation with right-hand side d = gcd $(a_1, a_2, \ldots, a_n)$ as well as the generator. Then a solution for (2) can be found as $(a_0/d)x^d$.

2. THE ALGORITHM

We now present the algorithm to find the general solution of the linear diophantine equation (2). The method is based on Lemma 1, namely, it finds the generator $(x_{F_1}, x_{F_2}, \ldots, x_{F_{n-1}})$ of (1) as well as any one solution x^0 of (2), so that any solution of (2) can be expressed as in (3). A solution x^0 of (2) is found as a by-product of finding the generator. We list the steps:

Step 0. Set $k = 1$, $b_1^{(1)} = a_1$, $b_2^{(1)} = a_2$, $\ldots$, $b_n^{(1)} = a_n$, and $N = n$.

Also let

$x(b_1^{(1)}) = (1, 0, \ldots, 0)$, $x(b_2^{(1)}) = (0, 1, 0, \ldots, 0)$, $\ldots$,

$x(b_n^{(1)}) = (0, \ldots, 0, 1)$,

where $x(b)$ denotes the solution of (2) with right-hand side $a_0 = b$.

Step 1. Find integers $\ell_2, \ell_3, \ldots, \ell_N$ so that they satisfy

$$\begin{aligned} b_1^{(k)} &= \ell_2 b_2^{(k)} + r_2, \quad 0 \le r_2 < b_2^{(k)} \\ r_2 &= \ell_3 b_3^{(k)} + r_3, \quad 0 \le r_3 < b_3^{(k)} \\ &\vdots \\ r_{N-1} &= \ell_N b_N^{(k)} + r_N, \quad 0 < r_N = b_1' < b_N^{(k)}, \end{aligned}$$

and thus

$$b_1^{(k)} = \ell_2 b_2^{(k)} + \ell_3 b_3^{(k)} + \cdots + \ell_N b_N^{(k)} + b_1'.$$

Step 2. Find a solution $x(b_1')$ for $a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = b_1'$ as follows:

$$x(b_1') = x(b_1^{(k)}) - \ell_2 x(b_2^{(k)}) - \ell_3 x(b_3^{(k)}) - \cdots - \ell_N x(b_N^{(k)}).$$

Step 3. If $b_1' = 0$, $x(b_1')$ is one of the generating vectors. Eliminate one variable, i.e., $N = N - 1$, and set

$$b_1^{(k+1)} = b_2^{(k)},\ b_2^{(k+1)} = b_3^{(k)},\ \ldots,\ b_N^{(k+1)} = b_{N+1}^{(k)}.$$

If $N = 1$, go to Step 4 (termination). If $N > 1$, increment the iteration count (i.e., $k = k + 1$) and return to Step 1.

If $b' \neq 0$, set $b_1^{(k+1)} = b_2^{(k)},\ b_2^{(k+1)} = b_3^{(k)},\ \ldots, b_N^{(k+1)} = b_1'$, $k = k + 1$ and return to Step 1.

Step 4. We now have $(n - 1)$ generating vectors for (1), and $b_1^{(k+1)}$ is the gcd $(a_1, a_2, \ldots, a_n)$. A solution for (2) can be found as

$$\frac{a_0}{b_1^{(k+1)}}\, x(b_1^{(k+1)}).$$

Stop.

We now give three results which show the validity of the algorithm.

Theorem 1: There is a one-to-one correspondence between the solutions of (4) and (5):

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = 0; \tag{4}$$

$$b_1^{(k)}y_1 + b_2^{(k)}y_2 + \cdots + b_n^{(k)}y_n = 0. \tag{5}$$

Here $b_1^{(k)} \neq 0$, $b_2^{(k)} \neq 0$, ..., $b_n^{(k)} \neq 0$ correspond to the values obtained for b_i in the kth iteration of Step 1, as far as $N = n$.

Proof: Consider the following two equations corresponding to any two consecutive iterations of the algorithm:

$$(k\text{th iteration})\quad b_1^{(k)}y_1^{(k)} + b_2^{(k)}y_2^{(k)} + \cdots + b_n^{(k)}y_n^{(k)} = 0;$$

$$(k+1\text{st iteration})\quad (b_1^{(k)} - \ell_2 b_2^{(k)} - \cdots - \ell_n b_n^{(k)})y_1^{(k+1)} + b_2^{(k)}y_2^{(k+1)} + \cdots + b_n^{(k)}y_n^{(k+1)} = 0.$$

The second equation can be written as

$$b_1^{(k)}y_1^{(k+1)} + b_2^{(k)}(y_2^{(k+1)} - \ell_2 y_1^{(k+1)}) + \cdots + b_n^{(k)}(y_n^{(k+1)} - \ell_n y_1^{(k+1)}) = 0.$$

This means

$$y_1^{(k)} = y_1^{(k+1)},\ y_2^{(k)} = y_2^{(k+1)} - \ell_2 y_1^{(k+1)},\ \ldots,\ y_n^{(k)} = y_n^{(k+1)} - \ell_n y_1^{(k+1)}.$$

Using vector-matrix notation, we have

$$y^{(k)} = \begin{pmatrix} y_1^{(k)} \\ y_2^{(k)} \\ \vdots \\ y_n^{(k)} \end{pmatrix} = \begin{pmatrix} -\ell_2 & 1 & 0 & 0 \\ -\ell_3 & 0 & 1 & 0 \\ \vdots & \vdots & & \vdots \\ -\ell_n & 0 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} y_1^{(k+1)} \\ y_2^{(k+1)} \\ \vdots \\ y_n^{(k+1)} \end{pmatrix} = Ty^{(k+1)}.$$

Notice that $|\det T|$ (i.e., the absolute value of the determinant of T) $= 1$. We now show inductively on k that there exists a matrix M such that

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = My^{(k)}$$

which satisfies $|\det M| = 1$. Clearly, for the first iteration, $T = M$ and

$$|\det T| = |\det M| = 1.$$

Assume that there exists a matrix M' with $|\det M'| = 1$ such that $x = M'y^{(k)}$. Substituting $y^{(k)} = Ty^{(k+1)}$, we get $x = M'Ty^{(k+1)}$. As

$$|\det (M'T)| = |\det M'| \times |\det T| = 1.$$

Thus, $x = My^{(k+1)}$ where $M = M'T$ and $|\det M| = 1$.

It is well known (e.g., see [7]) that, if there exists a matrix M such that $x = My$ with $|\det M| = 1$, there is a one-to-one correspondence between the solutions x and y. Thus, the theorem is proved. Q.E.D.

Theorem 2: If $(y_{F_1}, y_{F_2}, \ldots, y_{F_{n-1}})$ is the generator of (5), the corresponding $(x_{F_1}, x_{F_2}, \ldots, x_{F_{n-1}})$ is the generator of (4).

Proof: Assume that $(x_{F_1}, x_{F_2}, \ldots, x_{F_{n-1}})$ is not the generator of (4). Then there exists a solution vector x satisfying (4) such that it cannot be expressed as an integer linear combination of $x_{F_1}, x_{F_2}, \ldots, x_{F_{n-1}}$. However, because of the one-to-one correspondence (Theorem 1), there exists a unique y which corresponds to x (i.e., $My = x$), and there are integers $\beta_1, \beta_2, \ldots, \beta_{n-1}$ such that $y = \beta_1 y_{F_1} + \beta_2 y_{F_2} + \cdots + \beta_{n-1} y_{F_{n-1}}$ as $(y_{F_1}, y_{F_2}, \ldots, y_{F_{n-1}})$ is the generator. However,

$$\begin{aligned} x = My &= M(\beta_1 y_{F_1} + \beta_2 y_{F_2} + \cdots + \beta_{n-1} y_{F_{n-1}}) \\ &= \beta_1 My_{F_1} + \beta_2 My_{F_2} + \cdots + \beta_{n-1} My_{F_{n-1}} \\ &= \beta_1 x_{F_1} + \beta_2 x_{F_2} + \cdots + \beta_{n-1} x_{F_{n-1}}, \end{aligned}$$

and thus a contradiction. Q.E.D.

Theorem 3: Assume that $d = \gcd(a_1, a_2, \ldots, a_n) = \gcd(a_2, \ldots, a_n)$. Then the general solution of (6) can be expressed as $x = kx^0 + x'$, where k is an integer, x^0 any solution of (7), and x' the general solution of (8).

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = 0; \tag{6}$$

$$a_1 + a_2x_2 + \cdots + a_nx_n = 0; \tag{7}$$

$$a_2x_2 + \cdots + a_nx_n = 0. \tag{8}$$

Proof: Since d divides a_1, we have, for ℓ integer, $a_1 = \ell d$, and thus there are solutions $x_2, x_3, \ldots, x_n$ to (7). This means (6) has solutions when x_1 is fixed to any integer. Clearly, x^0 is any such solution to (6) in which $x_1 = 1$. Observe that all solutions for (6) can be characterized by fixing x_1 to any integer k and solving (6) in the remaining variables, $x_2, x_3, \ldots, x_n$. More specifically, for x_1 fixed to k, we want all solutions which satisfy

$$a_2x_2 + \cdots + a_nx_n = -a_1k. \tag{6}'$$

From Lemma 1, however, solutions for (6)' can be expressed as a sum of a solution for (6)' and the general solution for (8). Thus,

$$x = kx^0 + (k_1x_{F_1} + k_2x_{F_2} + \cdots + k_{n-2}x_{F_{n-2}})$$

is the general solution for (6) for integer, $k_1, k_2, \ldots, k_{n-2}$, where kx^0 is a solution satisfying (6)' and $(x_{F_1}, x_{F_2}, \ldots, x_{F_{n-2}})$ is the generator of (8) with $x_1 = 0$. Setting

$$x' = \sum_{i=1}^{n-2} k_i x_{F_i},$$

means that x' is any solution to (8), and hence the result. Q.E.D.

3. EXAMPLE AND DISCUSSION

Table 1 lists the computational process for finding the generator (x_{F_1}, x_{F_2}) for a 3-variable diophantine equation with the right-hand side equal to zero. The two vectors

$$\begin{pmatrix} -57 \\ 17 \\ 94 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 61 \\ -95 \\ -1 \end{pmatrix}$$

form the generator.

From Theorem 1, there is a one-to-one relationship between (9) and (10):

(9) $$8913x_1 + 5677x_2 + 4378x_3 = 0;$$

(10) $$10y_1 + 5y_2 + 3y_3 = 0.$$

The relationship is $x = My$, where

$$M = \begin{pmatrix} -3 & 27 & 4 \\ -3 & -10 & 13 \\ 10 & -42 & -25 \end{pmatrix}, \quad |\det M| = 1.$$

From Theorem 2, the generator (y_{F_1}, y_{F_2}) of (10), if found, will be translated to the generator

$$(x_{F_1}, x_{F_2}) = (My_{F_1}, My_{F_2})$$

of (9).

Iteration 10 of the algorithm (cf. Table 1) finds a solution

$$y = \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}$$

for (10), and from Theorem 3, the general solution for (10) can be found as

$$k\begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix} + y', \text{ where } y' = \begin{pmatrix} 0 \\ y_2 \\ y_3 \end{pmatrix}$$

is the general solution for (10) with $y_1 = 0$. Iterations 11 through 13 are performed to find the general solution for

(11) $$5y_2 + 3y_3 = 0.$$

It can easily be checked that the general solution for (11) is

$$y' = \ell\begin{pmatrix} 0 \\ 3 \\ -5 \end{pmatrix} \text{ for } \ell \text{ integer.}$$

Thus,

$$\begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 \\ 3 \\ -5 \end{pmatrix}$$

form a generator for (10).

TABLE 1. ALGORITHM COMPUTATIONS

Iteration k	$b_1^{(k)} = \ell_2 b_2^{(k)} + \ell_3 b_3^{(k)} + b_1'$	x_1	x_2	x_3
	8913	1	0	0
	5677	0	1	0
	4378	0	0	1
1	8913 = 1(5677) + 0(4378) + 3236	1	-1	0
2	5677 = 1(4378) + 0(3236) + 1299	0	1	-1
3	4378 = 1(3236) + 0(1299) + 1142	-1	1	1
4	3236 = 2(1299) + 0(1142) + 638	1	-3	2
5	1299 = 1(1142) + 0 (638) + 157	1	0	-2
6	1142 = 1 (638) + 3 (157) + 33	-5	4	5
7	638 = 4 (157) + 0 (33) + 10	-3	-3	10
8	157 = 4 (33) + 2 (10) + 5	27	-10	-42
9	33 = 3 (10) + 0 (5) + 3	4	13	-25
10	10 = 2 (5) + 0 (3) + 0	-57	17	94
11	5 = 1 (3) + 2	23	-23	-17
12	3 = 1 (2) + 1	-19	36	-8
13	2 = 2 (1) + 0	61	-95	-1

In general, whenever the final remainder (i.e., b') of Step 1 in each iteration becomes 0, we obtain a vector which is one of the $n - 1$ generating vectors, and the size of problem (i.e., the number of variables) is reduced by 1.

Theorem 3 shows that this elimination of one variable at a time guarantees the generating characteristic. After the problem is reduced, the same arguments (i.e., Theorem 1-Theorem 3) will be applied to the reduced problem, sequentially. Eventually, a 2-variable problem will be solved which yields the $(n - 1)$st or last generating vector, and the process terminates.

From Lemmas 2 and 3, the last nonzero remainder in the algorithm gives the greatest common divisor of $a_1, a_2, \ldots, a_n$. In the example, detailed in Table 1, the last nonzero remainder is 1 and is the gcd of 8913, 5677, and 4378. To see this, note that

$$\gcd(8913, 5677, 4378) = \gcd(10, 5, 3)$$

by Lemma 2 which, in turn, is equal to gcd (5, 3) by Lemma 3. Repeating the same argument gives

$$\gcd(5, 3) = \gcd(3, 2) = \gcd(2, 1) = \gcd(1) = 1,$$

or

$$\gcd(8913, 5677, 4378) = 1.$$

Finally, Table 1 displays a solution for the equation with the right-hand side equal to 1 = gcd (8913, 5677, 4378). The general solution for the equation with the right-hand side a_0 can then be expressed as:

$$a_0 \begin{pmatrix} -19 \\ 36 \\ -8 \end{pmatrix} + k_1 \begin{pmatrix} -57 \\ 17 \\ 94 \end{pmatrix} + k_2 \begin{pmatrix} 61 \\ -95 \\ -1 \end{pmatrix},$$

where k_1 and k_2 are integers.

REMARKS

1. An examination of the algorithm indicates that the divisions in Step 1 can be made computationally more efficient by using the least absolute remainder rather than the positive remainder. Specifically, we find ℓ_i (i = 2, ..., N) such that $|r_i|$ is minimized ($0 \leq |r_i| \leq b_i^{(k)}$) in Step 1, rather than using r_i, where $0 \leq r_i \leq b_i$. This change allows the proofs of the theorems to go through essentially unchanged.

2. The preceding discussion can be used to show that the Blankinship Alsorithm [1] for finding the gcd of n integers will also find the general solution of a linear diophantine equation. Specifically, the algorithm presented here can be regarded as a modified Blankinship Algorithm where the modification is in selecting the operators (according to Blankinship's terminology). The Blankinship Algorithm, on the other hand, can be regarded as a special case of our method where $\ell_2 \equiv \ell_3 \equiv \ldots \equiv \ell_{n-1} \equiv 0$ in Step 2 of the algorithm presented here.

REFERENCES

1. W. A. Blankinship. "A New Version of the Euclidean Algorithm." *American Math. Monthly* 70, No. 7 (1963):742-745.
2. R. Weinstock. "Greatest Common Divisor of Several Integers and an Associated Linear Diophantine Equation." *American Math. Monthly* 67, No. 7 (1960):664-667.
3. J. Bond. "Calculating the General Solution of a Linear Diophantine Equation." *American Math. Monthly* 74, No. 8 (1967):955-957.
4. W. Chase. "The Indirect Threat Algorithm." Technical Memorandum, Naval Electronics Laboratory Center, San Diego, California, November 1975.
5. S. Morito & H. M. Salkin. "A Comparison of Two Heuristic Algorithms for a Radio Frequency Intermodulation Problem." Technical Memorandum, Case Western Reserve University, Department of Operations Research, Cleveland, Ohio, October 1977.
6. T. Saaty. *Optimization in Integers and Related Extremal Problems*. New York: McGraw-Hill, 1970.
7. H. M. Salkin. *Integer Programming*. Reading, Mass.: Addison-Wesley, 1975.

*****