

Integer Sets with Distinct Subset-Sums

By W.F. Lunnon

Abstract. In Section 1 we introduce the problem of finding minimal-height sets of n natural numbers with distinct subset-sums (SSD), and in Section 2 review the well-known Conway-Guy sequence \mathbf{u} , conjectured to yield a minimal SSD set for every n . We go on (Section 3) to prove that \mathbf{u} certainly cannot be improved upon by any "greedy" sequence, to verify numerically (Section 4) that it does yield SSD sets for $n < 80$, and (Section 5) by direct search to show that these are minimal for $n \leq 8$. There is a brief interlude (Section 6) on the problem of decoding the subset from its sum. In Section 7 generalizations of \mathbf{u} are constructed which are asymptotically smaller: Defining the *Limit Ratio* of a sequence \mathbf{w} to be $\alpha = \lim_{n \rightarrow \infty} w_n/2^{n-1}$, the Atkinson-Negro-Santoro sequence \mathbf{v} (known to give SSD sets) has $\alpha = 0.6334$, Conway-Guy (conjectured to) has $\alpha = 0.4703$, and our best generalization has $\alpha = 0.4419$. We also (Section 8) discuss when such sequences have the same α , and (Section 9) how α may efficiently be computed to high accuracy.

1. The Distinct Subset-Sum Problem. A well-known problem in combinatorial number theory [2, pp. 64-65] involves the construction for given n of a set $\mathbf{p} = \{p_i\}$, $i = 1, \dots, n$, of natural numbers possessing the property we shall call *Subset-Sum Distinctness*, or SSD for short: That is,

$$(1.1) \quad \sum_{i \in S} p_i = \sum_{i \in T} p_i \Rightarrow S = T \subseteq \{1, \dots, n\},$$

or, distinct subsets of \mathbf{p} have distinct sums. Evidently, choosing $p_i = 2^{i-1}$ satisfies (1.1); the interest lies in how much the maximum element p_n can be reduced below 2^{n-1} .

In order to explore the SSD property, it is convenient to consider a more general concept: We say that x has a *representation* by \mathbf{p} , with length k and signature l , when for some S and T as before

$$(1.2) \quad x = \sum_{i \in S} p_i - \sum_{i \in T} p_i,$$

where $S \cap T = \emptyset$, $|S| + |T| = k$, $|S| - |T| = l$; that is, x is the difference of two disjoint subset-sums, of which the first has l terms more than the second, and the two together have k terms. Alternatively,

$$(1.3) \quad x = \sum_{i=1}^n e_i p_i,$$

Received June 23, 1986; revised October 20, 1986.

1980 *Mathematics Subject Classification* (1985 Revision). Primary 11B13, 05-04, 05A99, 11J13, 94A60, 65B05.

Key words and phrases. SSD, Conway-Guy, Knapsack problem, backtrack search, divide-and-conquer, Richardson extrapolation, MACSYMA, MDCF.

where $e_i \in \{-1, 0, +1\}$, $\sum e_i = l$, $\sum |e_i| = k$. In terms of this notation, \mathbf{p} is SSD when there is no nonempty representation of zero.

Interesting sets turn out to be associated with certain infinite sequences $\mathbf{w} = (w_n)$, $n = 0, 1, 2, \dots$, in the following fashion: Given \mathbf{w} and n , set

$$(1.4) \quad p_i = w_n - w_{n-i}, \quad i = 1, \dots, n.$$

The significance of this definition is that the n th set \mathbf{p}^n is obtained from \mathbf{p}^{n-1} by attaching a zero, then incrementing every element by the same quantity $w_n - w_{n-1}$. Evidently, the maximum p_n equals w_n .

One such sequence is the *Atkinson-Negro-Santoro* sequence \mathbf{v} [12], whose first few values are

$$(1.5) \quad \begin{array}{cccccccccccccccc} n & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ v_n & 0 & 1 & 2 & 4 & 7 & 13 & 24 & 46 & 88 & 172 & 337 & 667 & 1321 \end{array}$$

and which is defined by the recurrence

$$(1.6) \quad v_0 = 0, \quad v_1 = 1, \quad v_{n+1} = 2v_n - v_{n-m} \quad \text{for } n \geq 1,$$

where $m = \lfloor \frac{1}{2}n + 1 \rfloor$ is the greatest integer not exceeding $\frac{1}{2}n + 1$.

Example. If we take $n = 6$ in (1.4), from (1.5) we get the SSD set

$$(1.7) \quad \mathbf{p} = \{11, 17, 20, 22, 23, 24\},$$

which is the unique, optimal solution to the recreational problem posed in [3, no. 5].

(1.8) THEOREM. *Relation (1.4) produces a SSD set \mathbf{p} from \mathbf{v} for all n .*

Proof. In fact, \mathbf{p} has the further property that *only positive numbers x can be represented by positive signatures*. This will plainly be true if it is true for the largest T and smallest (longer, disjoint) S possible, that is, if

$$\sum_{i=1}^{\lfloor \frac{1}{2}n - \frac{1}{2} \rfloor} p_i - \sum_{i=\lfloor \frac{1}{2}n + 1 \frac{1}{2} \rfloor}^n p_i \geq 1.$$

Rewriting in terms of \mathbf{v} , and remarking that equality gives us the smallest possible sequence, we get

$$(1.9) \quad v_n = \sum_{i=\lfloor \frac{1}{2}n \rfloor}^{n-1} v_i - \sum_{i=0}^{\lfloor \frac{1}{2}n - 1 \frac{1}{2} \rfloor} v_i + 1.$$

This is easily seen to hold for the sequence (1.6). Now suppose the theorem to be false, and n to be the first for which it fails. By the foregoing, \mathbf{p} must represent zero with signature 0, that is $\sum_{i \in S} v_i = \sum_{i \in T} v_i$ where $|S| = |T|$, $0 \notin S$ (say), and $n \notin S \cup T$. But then $\{v_{n-1} - v_{n-1-i}\}$ would not be SSD, contrary to hypothesis. \square

Dividing by 2^{n-1} and iterating (1.6), it may further be shown that \mathbf{v} possesses a *limit ratio*,

$$(1.10) \quad v_n / 2^{n-1} \rightarrow \alpha_v \quad \text{where } \alpha_v = 0.63336835 \dots;$$

the computation of this and similar constants is pursued in Section 9. Finally, observe that if we have *any* particular SSD set \mathbf{p} of size n_0 and its corresponding \mathbf{w} , then for $n > n_0$ Eq. (1.9) of the construction above may be used to extend \mathbf{w} indefinitely, yielding arbitrarily large SSD sets. The resulting limit ratio will indeed

be smaller than at n_0 , although the improvement (of order $2^{-\frac{1}{2}n_0}$) is in practice negligible.

Rather more intriguing is the *Conway-Guy* sequence \mathbf{u} , whose first few values are

$$(1.11) \quad \begin{array}{cccccccccccccccc} n & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ m & - & 1 & 2 & 2 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 5 & 5 \\ u_n & 0 & 1 & 2 & 4 & 7 & 13 & 24 & 44 & 84 & 161 & 309 & 594 & 1164 \end{array}$$

and which may be defined by

$$(1.12) \quad u_0 = 0, \quad u_1 = 1, \quad u_{n+1} = 2u_n - u_{n-m} \quad \text{for } n \geq 1,$$

where $m = \lceil \frac{1}{2} + \sqrt{2n} \rceil$. This choice of m ensures that $T_{m-1} < n \leq T_m$ for $n > 0$, where

$$(1.13) \quad T_m = \frac{1}{2}m(m+1)$$

is the m th triangular number.

In [1] is advanced the following

(1.14) CONJECTURE. *Relation (1.4) produces a SSD set from \mathbf{u} for all n . We christen this the Conway-Guy conjecture, and Sections 3–5 comprise some investigations into the matter. The limit ratio of \mathbf{u} is*

$$u_n/2^{n-1} \rightarrow \alpha_{\mathbf{u}} \quad \text{where } \alpha_{\mathbf{u}} = 0.47025057 \dots$$

Finally, in [1] and [2, pp. 64–65] is further proposed the

(1.15) CONJECTURE. \mathbf{u} and $\alpha_{\mathbf{u}}$ are “in essence” best possible; though the author appears to have something relatively weak in mind, we choose to interpret this as doubting whether sets with smaller α are possible. In Section 7 we shall consider generalizations of \mathbf{u} which contradict (1.15), whether or not they also satisfy (1.14).

2. The Conway-Guy Sequence. There follows a more succinct presentation of the material in Theorems 5–8 of [1]. We assume implicitly $n \geq 0$, and u_n and m defined as in (1.12). Summation is implicitly over i .

$$(2.1) \text{ LEMMA. } \textit{We have } u_n \geq 0, \quad u_{n+1} - u_n > 0, \quad f_n = u_{n+2} - u_{n+1} - u_n > 0.$$

Proof. By induction using (1.12): For the last part, note that $f_0 = 1$, and for $n \geq 2$,

$$\begin{aligned} \Delta f_{n-2} &= u_{n+1} - 2u_n + u_{n-2} \\ &= u_{n-2} - u_{n-m} \quad \text{by (1.12)} \\ &\geq 0 \quad \text{by the previous parts, noting } m \geq 2 \text{ if } n \geq 2. \quad \square \end{aligned}$$

If a sequence \mathbf{w} has no nonempty representation of zero with signature 0, we christen it *SSD0*. It turns out that this property is equivalent to the associated sets \mathbf{p} being SSD, at any rate when \mathbf{w} is the Conway-Guy sequence: for

(2.2) THEOREM. *Let the set \mathbf{p} be associated with the sequence $\mathbf{w} = \mathbf{u}$ by (1.4) for some fixed n . Then \mathbf{p} is SSD when \mathbf{u} is SSD0.*

Proof. Suppose $\sum_S p_i = \sum_T p_i$, with $|S| - |T| = l \geq 0$, say. Then by (1.4),

$$(2.3) \quad u_n l - \sum_S u_{n-i} + \sum_T u_{n-i} = 0.$$

Taking the greatest possible number of large terms in the first sum and small in the second, the left-hand side of this is

$$\geq u_n l - \sum_{[\frac{1}{2}(n-l+1)]}^{n-1} u_i + \sum_0^{[\frac{1}{2}(n-l-2)]} u_i = g_n, \quad \text{say.}$$

But $g_l = \sum_0^{l-1} (u_l - u_i) > 0$ by (2.1), provided $l > 1$; and for $n > l$,

$$\begin{aligned} \Delta g_n &= l(u_{n+1} - u_n) - u_n + u_{[\frac{1}{2}(n-l+2)]} \\ (2.4) \quad &= (l-2)(u_{n+1} - u_n) + 2u_{n+1} - 3u_n + u_{[\frac{1}{2}(n-l+2)]} \\ &= (l-2)(u_{n+1} - u_n) + u_n - 2u_{n-m} + u_{[\frac{1}{2}(n-l+2)]} \\ &> 0, \quad \text{provided } l \geq 2 \text{ and } m \geq 2 \text{ by (2.1).} \end{aligned}$$

In fact, if $n \geq l \geq 2$, then we must have $m \geq 2$, so (2.3) is impossible unless $l = 0, 1$. If $l = 1$ we can make it 0 simply by attaching $i = n$ to T ; so finally $l = 0$. The argument is easily reversible. \square

Notice that (2.2) would hold for any \mathbf{w} provided only that it satisfied the relatively weak condition $g_l > 0$ for $l > 1$ established above (2.4) for \mathbf{u} .

As it happens, there are rather a lot of representations of zero by \mathbf{u} with signatures other than 0. With T_m as in (1.13), we have

LEMMA. If $T_{m-1} \leq n \leq T_m$ then

$$(2.5) \quad 1 + \sum_0^{m-1} u_{n-i} = u_{n+1} + \sum_0^{m-1} u_{T_{i-1}}.$$

Proof. Observe that Eq. (2.5) is immediately equivalent to (1.12), then invoke the uniqueness of \mathbf{u} . [Notice that $n = T_{m-1}$ is included here, in contrast to (1.12): In this instance the terms $i = m-1$ cancel from both sides, reducing to the previous situation.] \square

For $m \geq 3$ the 1 and the u_{T_2} cancel from both sides, giving representations of zero whose size $2m-2$ increases with n . For fixed size the search for a counterexample to property SSD0 is indeed effectively bounded by

(2.6) THEOREM. If there is a representation of zero with signature 0 and size $2k$ by (u_0, \dots, u_n) , then we can take $n = T_k + 1$.

Proof. Let u_{n+1} be the largest term occurring in such a representation, and suppose the corresponding $m \geq 3$. Then

$$\begin{aligned} u_{n+1} &= \sum_0^{m-1} u_{n-i} + 1 - \sum_0^{m-1} u_{T_{i-1}} \quad \text{by (2.5)} \\ &= \sum_0^{m-2} u_{n-i} + u_{n-m+1} - \sum_3^{m-1} u_{T_{i-1}} \\ &\geq \sum_0^{m-2} u_{n-i} + \left[u_{T_{m-2}+1} - \sum_3^{m-1} u_{T_{i-1}} \right]. \end{aligned}$$

If $m = 3$, the bracketed quantity $[\dots] = u_2 > 0$; and if $m > 3$,

$$\Delta_m[\dots] = u_{T_{m-2}+1} - u_{T_{m-2}} - u_{T_{m-3}+1} > 0 \quad \text{by (2.1).}$$

Therefore,

$$u_{n+1} > \sum_0^{m-2} u_{n-i};$$

and since the right-hand side is the maximum possible sum of $m - 1$ terms whose largest is u_n , it follows that the size $2k$ of the representation is at least $2m$; that is, given k , the maximum possible value of $n + 1$ is $T_k + 1$. The cases $m < 3$, i.e., $n + 1 \leq 4$, are easily disposed of by inspection. \square

3. The Local Optimality of u . As we have remarked, it is not known that u is SSD0 for all n . However, we can show that no number smaller than u_n may be attached to the previous members of the sequence. In pursuit of this end, we define a pair of functions $a = a_{nl}$, $b = b_{nl}$, such that the entire open interval (b, a) turns out to be representable by $\{u_0, \dots, u_{n-1}\}$ with signature l . [For this purpose we include the empty representation of zero, with $k = l = 0$.] As before, $T_{m-1} < n \leq T_m$, and summation is over i . We make use of a *directed sum* convention, similar to that for integrals: $\sum_p^{q-1} = -\sum_q^{p-1}$.

Definition. For $|l| \leq m$,

$$(3.1) \quad a_{nl} = u_{n-l+1} + \sum_1^{l-1} u_{n-i}; \quad b_{nl} = -u_{n+l+1} + \sum_1^{l+1} u_{n+l+1-i}.$$

Each of these is equivalent to the other, by virtue of directed sums and the symmetry relation

$$(3.2) \quad b_{nl} = -a_{n,-l}$$

following from the obvious fact that, when x is representable with signature l , then $-x$ is also with $-l$.

Below in (3.3) is a short table of (b_{nl}, a_{nl}) , for $l \geq 0$ only, in view of (3.2).

n/l	0	1	2	3
0	(-1, 1)			
1	(-1, 1)	(-1, 1)		
2	(-2, 2)	(-1, 2)	(0, 2)	
3	(-3, 3)	(-2, 4)	(0, 4)	
4	(-6, 6)	(-4, 7)	(0, 8)	(4, 8)
5	(-11, 11)	(-7, 13)	(-3, 14)	(4, 15)
6	(-20, 20)	(-16, 24)	(-9, 26)	(4, 27)

(3.4) LEMMA. *The endpoints of the interval are consistent, that is, $a_{nl} > b_{nl}$.*

Proof. Relation (3.2) takes care of $l < 0$, so we assume $0 \leq l \leq m$. Then,

$$\begin{aligned} a_{nl} &= u_{n-l+1} + \sum_1^{l-1} u_{n-i} \quad \text{by (3.1)} \\ &= u_{n-l+1} - u_n + \sum_0^{l-1} u_{n-i} \\ &= u_{n+1} - u_n + u_{n-l+1} - 1 - \sum_l^{m-1} u_{n-i} - \sum_0^{m-1} u_{T_{i-1}} \end{aligned}$$

by (2.5) and rearranging.

In a similar fashion,

$$\begin{aligned}
 b_{nl} &= -u_{n+l+1} + \sum_1^{l+1} u_{n+l+1-i} \\
 &= -u_{n+l+1} + u_n + \sum_0^{l-1} u_{n+l-i} \\
 &= u_n - 1 - \sum_l^{m'-1} u_{n+l-i} + \sum_0^{m'-1} u_{T_{l-1}},
 \end{aligned}$$

where m' is chosen so that $T_{m'-1} < n + l \leq T_{m'}$. There are now two cases to consider: Since $n \leq n + l \leq n + m$ and $T_{m-1} + m = T_m$, either $m' = m$ or $m' = m + 1$. In the first case,

$$\begin{aligned}
 a_{nl} - b_{nl} &= [u_{n+1} - 2u_n + u_{n-l+1}] + \sum_l^{m-1} [u_{n+l-i} - u_{n-i}] \\
 &= [u_{n-l+1} - u_{n-m}] + \sum_l^{m-1} [u_{n+l-i} - u_{n-i}] \\
 &> 0 \quad \text{by (2.1), the bracketed terms being nonnegative.}
 \end{aligned}$$

In the second case, the right-hand side contains the extra term $[u_{n+l-m} - u_{T_{m-1}}]$, which is nonnegative since $n + l - m > T_m - m = T_{m-1}$. \square

(3.5) COROLLARY. *Setting $n = T_m - 1$, $l = m$ in the above expression (second case), we have in this instance $a_{nl} - b_{nl} = u_{T_m-1} - u_{(T_m-1)} = u_{T_m} - u_{T_m-1}$ by (1.12).*

LEMMA. *Adjacent intervals "overlap" for the purposes of (3.8) in the sense that*

$$(3.6) \quad a_{n,l-1} + u_n \geq a_{nl} > b_{n,l-1} + u_n \geq b_{nl} \quad \text{for } l > -m,$$

$$(3.7) \quad a_{nl} \geq a_{n,l+1} - u_n > b_{nl} \geq b_{n,l+1} - u_n \quad \text{for } l < m.$$

Proof. First observe that (3.7) is obtained from (3.6) by substituting $l + 1$ for l and subtracting u_n . For the first inequality of (3.6),

$$\begin{aligned}
 a_{n,l-1} + u_n - a_{nl} &= u_{n-l+2} + \sum_1^{l-2} u_{n-i} + u_n - u_{n-l+1} - \sum_1^{l-1} u_{n-i} \quad \text{by (3.1)} \\
 &= u_{n-l+2} - 2u_{n-l+1} + u_n \\
 &= \begin{cases} u_n - u_{n-m} & \text{if } T_{m-1} < n - l + 1, \\ u_n - u_{n-m+1} & \text{if } T_{m-1} \geq n - l + 1 \end{cases} \\
 &\geq 0.
 \end{aligned}$$

The third is proved similarly, or we can use (3.2) and directed sums. The second is attacked along the lines of (3.4): To begin with, we assume that $l \geq 0$, and $n + l - 1 \leq T_m$. Then

$$\begin{aligned}
 a_{nl} - b_{n,l-1} - u_n &= u_{n-l+1} + [u_{n+l-m} - u_{n-m}] + \sum_l^{m-1} [u_{n+l-i} - u_{n-i}] \\
 &> 0 \quad \text{since } 0 \leq l \leq m.
 \end{aligned}$$

As in the proof of (3.4), if instead $n + l - 1 > T_m$ then the right-hand side gains the extra nonnegative term $[u_{n+l-1-m} - u_{T_{m-1}}]$. Finally, if $-m < l < 0$ we can use (3.2) to transform (3.6) into the equivalent (3.7) with $l > 0$. \square

LEMMA. For $0 \leq l \leq m$, (b, a) satisfies the recursion

$$(3.8) \quad \begin{aligned} a_{00} &= -b_{00} = 1, \\ a_{n+1,l} &= a_{n,l-1} + u_n, \\ b_{n+1,l} &= \begin{cases} b_{n,l-1} + u_n & \text{if } T_{l-1} = n, \\ b_{nl} & \text{if } T_{l-1} < n \leq T_l, \\ b_{n,l+1} - u_n & \text{if } T_l < n. \end{cases} \end{aligned}$$

For $-m \leq l < 0$ we may apply (3.2) to the above.

Proof. The first, second and fifth lines are immediate from Definition (3.1). For the others, we transform (3.1) via (2.5) as before to get

$$b_{nm} = \sum_0^{m-1} u_{T_{i-1}} - 1;$$

for the fourth line, m is the same when n is replaced by $n + 1$; for the third line, m increases by 1 and the extra term required on the right-hand side is just $u_{T_{m-1}} = u_n$. \square

(3.9) THEOREM. For $b_{nl} < x < a_{nl}$, x is representable with signature l by $\{u_0, \dots, u_{n-1}\}$.

Proof. By induction on n . From definition (1.3), x is representable with signature l by the larger set $\{u_0, \dots, u_n\}$ when there is some y representable by $\{u_0, \dots, u_{n-1}\}$ such that either (i) $x = y + u_n$ and y has signature $l - 1$; or (ii) $x = y$ and y has signature l ; or (iii) $x = y - u_n$ and y has signature $l + 1$. If the theorem is assumed for all l and some given n , we can build up a consecutive interval (b, a) of such x from some combination of these three cases, provided the intervals in question exist, (3.4), and are sufficiently close to one another, (3.6). For example, provided $l < m$ (so that $n > T_{l+1}$), by (3.4) all three smaller intervals exist, and by (3.6) the interval $(b_{n,l-1}, a_{n,l-1}) + u_n$ includes the top endpoint of (b_{nl}, a_{nl}) , and by (3.7) (b_{nl}, a_{nl}) includes the top of $(b_{n,l+1}, a_{n,l+1}) - u_n$; therefore their union $(b_{n,l+1} - u_n, a_{n,l-1} + u_n)$ is representable, and by (3.8) that is exactly $(b_{n+1,l}, a_{n+1,l})$. The various special cases for $l = m$ may be verified similarly. Finally, at $n = 0$ we have $(b_{00}, a_{00}) = (-1, +1)$ by (3.3) or (3.8), which is correct since zero is representable by the null set. \square

(3.10) LEMMA. For $n > 0$, $a_{n1} = u_n$ and $b_{n1} < 0$.

Proof. Use (3.1), (2.1). \square

(3.11) THEOREM. The set $\{u_0, \dots, u_{n-1}, x\}$ fails to be SSD0 if $x < u_n$.

Proof. By (3.9), (3.10), if $0 \leq x < u_n$ then x is representable with signature 1 by $\{u_0, \dots, u_{n-1}\}$, say

$$x = \sum_{i \in S} u_i - \sum_{i \in T} u_i \quad \text{where } |S| - |T| = 1.$$

So

$$0 = \sum_{i \in S} u_i - \left(x + \sum_{i \in T} u_i \right)$$

is a representation of zero with signature 0 by the extended set. \square

The natural way to present the results of this section would be to start from the conditions (3.4), (3.6), (3.8) on (b, a) , then derive from the formulae (3.1) as a consequence, thus avoiding the “rabbit from a hat” atmosphere surrounding our present treatment; a similar criticism can be leveled at the very definition (1.12) of u_n , which structurally ought to follow from the final theorem (3.11). However, the exigency of mathematical proof appears to preclude this intuitive approach, which is unfortunate for more than merely pedagogic reasons since it hampers the investigation of the generalized Conway-Guy sequences discussed in Section 7.

Another direction in which (3.11) may be extended is to consider a *spectrum* vector of exceptional values $(\dots, b^2, b^1 = b, a = a^1, a^2, \dots)$ which are not known to be representable with given n, l , rather than just the pair (b, a) dealt with previously. A useful subsidiary quantity in this connection is

$$(3.12) \quad t_{nj} = \sum_{i=1}^{j-1} [u_{T_{(m-1-i)}} - u_{T_{(m-1-i)}-1}] \quad \text{for } j = 1, \dots, m-1,$$

where $m = [\frac{1}{2} + \sqrt{2n}]$ as before. In terms of this quantity, the extended result is

(3.13) THEOREM. *If x is small and $\{u_0, \dots, u_{n-1}, x\}$ is SSD0, then x belongs to the spectrum set $\{u_n + t_{nj}\}$, the range of j being as above. [Here “small” means not exceeding the largest spectrum element $u_n + t_{n, m-1}$.]*

For example, for $n = 11$, $m = 5$, the four possible small candidates $x \leq 608$ for attachment to $\{0, 1, \dots, 309\}$ are

$$594 = u_{11}, \quad 605 = 594 + u_6 - u_5, \quad 607 = 605 + u_3 - u_2, \quad 608 = 607 + u_1 - u_0,$$

all of which do in practice yield SSD0 sets. A natural extension to the first Conway-Guy conjecture is to suggest that all the sets (3.13) are SSD0 for any n .

The proof of (3.13) is tedious rather than particularly difficult, most of the hard work having already been done above; we content ourselves with sketching the ideas. The vector version of the definition (3.1) of (b, a) is

$$(3.14) \quad a_{nl}^j = a_{nl} + t_{n-l+1, j}, \quad b_{nl}^j = b_{nl} + t_{n+l+1, j},$$

where j runs from 1 to the m -value (less unity) appropriate to $n-l$, $n+l+1$, respectively. The recursion turns out to run more smoothly if we extend the domain of definition (3.1) to include also

$$b_{nl} = -a_{n, -l} = a_{n-1, l-1} + u_{n-1} \quad \text{for } n = T_{l-1} \text{ only;}$$

otherwise, the n - and l -domains of both versions are the same. The symmetry lemma (3.2) becomes

$$(3.15) \quad b_{nl}^j = -a_{n, -l}^j \quad \text{for } j \text{ in range;}$$

the induction lemma (3.8) becomes, for $l \geq 0$,

$$\begin{aligned}
 (b_{00}^1, a_{00}^1) &= (-1, +1), \\
 (b_{01}^1, a_{01}^1) &= (-1, 0), \\
 (3.16) \quad a_{n+1,l}^j &= a_{n,l-1}^j + u_n \quad \text{for } n > T_l, \\
 b_{n+1,l}^j &= \begin{cases} b_{n,l-1}^{j-1} + u_n & \text{if } n = T_{l-1} - 1 \text{ and } j > 1, \\ a_{n,l-1}^1 + u_n & \text{if } n = T_{l-1} \text{ and } j = 1, \\ b_{nl}^j & \text{if } T_{l-1} \leq n \leq T_l, \\ b_{n,l+1}^j - u_n & \text{if } T_l < n. \end{cases}
 \end{aligned}$$

The only nontrivial idea in this intimidating concoction lies concealed in the case $n = T_{l-1} - 1$. Here a fresh spectrum, consisting only of b 's, is constructed inductively (with respect to n), by shifting one place to the left (with respect to l). As a result, all new intervals created are actually old (b, a) -style intervals, detached by shifting at some previous n and subsequently merely translated, without any further merging taking place. Consistency (3.4) and overlapping (3.6) are now trivial, and we arrive at the vector version of Theorem (3.9):

THEOREM. *Given a value x such that $b_{nl}^{m''} \leq x \leq a_{nl}^{m'}$, for x not to be representable by $\{u_0, \dots, u_{n-1}\}$ with signature l , it is necessary that x lies in the spectrum set*

$$(3.17) \quad \{b_{nl}^{j''}\} \cup \{a_{nl}^{j'}\},$$

where

$$1 \leq j' < m' = \left\lfloor \frac{1}{2} + \sqrt{2(n-l+1)} \right\rfloor, \quad 1 \leq j'' < m'' = \left\lfloor \frac{1}{2} + \sqrt{2(n+l+1)} \right\rfloor.$$

Proof. Where a new interval is created (at $n = T_{l-1}$), the translation

$$-(a_{n-1,l-1} - b_{n-1,l-1})$$

is added to all existing intervals (b^j 's only, if $l \geq 0$). By (3.5), this reduces to $-(u_{T_{m-1}} - u_{T_{m-1}-1})$; and the inductive accumulation of these translations, coupled with symmetry (3.15), leads to the formulae of (3.14). \square

Finally, (3.13) follows straightforwardly from (3.17) just as does (3.11) from (3.9).

4. Numerical Verification of SSD Property. The algorithms described below make heavy use of *backtracking*, a process during which a vector \mathbf{e} is made to take every value in turn from some predetermined set, while we search for a value with some particular property [4, Chapter 30]. In the present situation particularly, it is fruitful to regard the components of the vector as the individual digits of a *number*, written in some exotic number base: The value x of the number is then given by the scalar product $x = \mathbf{e} \cdot \mathbf{p}$, where \mathbf{p} is some fixed vector defining the values of the digit positions, and the property sought is that x should take some given subset of values. As x counts systematically through each possible value, the most-significant (high-index) digits of \mathbf{e} will vary most slowly, while the least-significant (low-index) vary most rapidly. *Impasse-avoidance* involves choosing the most-significant digit values so as to avoid partial vectors which can never lead to any desired x .

An obvious algorithm for the verification of the SSD property, given a set \mathbf{p} (now regarded as a vector with n components in ascending order of magnitude), runs as follows:

- (4.1) **ALGORITHM.** *Maintain a binary flag vector $f(x)$, setting $f(x) = 1$ initially for all x . Let \mathbf{e} backtrack over all binary vectors, and for each value $x = \mathbf{e} \cdot \mathbf{p}$ in turn set $f(x) = 0$; unless it is already zero, in which case we have detected a collision and \mathbf{p} cannot be SSD. Otherwise, on termination, \mathbf{p} is SSD.*

Since it essentially backtracks over all binary n -digit numbers, the time and space costs of this algorithm are plainly both of order $O(2^n)$; we need discuss it no further.

An improvement is afforded by the classical *divide-and-conquer* paradigm:

- (4.2) **ALGORITHM.** *Partition \mathbf{p} into $\mathbf{p}^1 = (p_1, \dots, p_{\lfloor \frac{1}{2}n \rfloor})$, $\mathbf{p}^2 = (p_{\lfloor \frac{1}{2}n \rfloor + 1}, \dots, p_n)$, and generate lists $\{x\}$ and $\{y\}$ of (natural) numbers represented by \mathbf{p}^1 and \mathbf{p}^2 , respectively, in the sense of (1.2). The two lists (which incidentally will contain many repetitions) are sorted separately, and then compared; when they contain no common value (a collision whose difference would represent zero), \mathbf{p} is SSD. [It is necessary to ignore a collision between empty representations of zero.]*

Generation of representations by an n -vector is essentially equivalent to backtracking through all ternary n -digit numbers \mathbf{e} , so the space cost is $s = O(\sqrt{3}^n)$, and the time (for sorting, [4, Chapters 8 and 9]) $s \log s = O(n\sqrt{3}^n)$.

For the sets \mathbf{p} discussed in Section 3, we could instead utilize (2.2) and test $\mathbf{u} = (u_0, \dots, u_n)$ for SSD0-hood. At first sight, this is scarcely an improvement: Since we are now only interested in collisions between the values of representations with *equal* signature l when using the divide-and-conquer approach, the backtracker must generate, separately for each l , only those $n + 1$ -digit ternary numbers whose digit-sum is exactly l . (The construction of an efficient algorithm to achieve this last is an unexpectedly diverting exercise!) It is true that the sets of values generated are now somewhat smaller than before, though in practice most of the time and space requirement is concentrated around $l = n/4$.

More importantly though, \mathbf{u} has a convenient property not shared by \mathbf{p} : Its elements are of widely differing sizes, indeed by (1.11) very roughly $u_i = O(2^i)$. This suggests the new

- (4.3) **ALGORITHM.** *Generate the set $\{y\}$ represented by the larger half \mathbf{u}^2 of \mathbf{u} , while immediately discarding any value y outside the interval (x_{\min}, x_{\max}) of values representable by the smaller half \mathbf{u}^1 . We may expect to be left with a very short list $\{y'\}$, which is now sorted alone; then as each value x represented by \mathbf{u}^1 is generated, a collision with x is immediately sought among the $\{y'\}$.*

This modification reduces the space requirement substantially: Experimentally, it now appears to be roughly $O(\beta\gamma^n)$, where $\beta = 10$ and $\gamma = 1.07$.

A further refinement is to utilize *impassé-avoidance* in the backtracker. Let $\mathbf{w} = (w_1, \dots, w_m)$ denote the set of base elements (e.g., \mathbf{u}^2) and $\{t\}$ the set of target-values (e.g., $\{y'\}$) to be represented with signature l . Define

$$c_{ij} = \sum_{k=\lfloor \frac{1}{2}(m-j+3) \rfloor}^m w_k - \sum_{k=1}^{\lfloor \frac{1}{2}(m-j) \rfloor} w_k, \quad d_{ij} = -c_{j,-i},$$

to be the largest and smallest values which are representable with signature j by the i least-significant base elements (w_1, \dots, w_i) ; and let

$$z_i = \sum_{k=i}^m w_k e_k$$

be the (partial) value represented by the current most-significant end of the (ternary) backtrack vector \mathbf{e} .

(4.4) **ALGORITHM.** *Suppose the backtracker has just incremented e_i . If none of the t lies within the current interval $(z_i + d_{i-1,l-j}, z_i + c_{i-1,l-j})$ of reachable values, then it would be futile to proceed further down \mathbf{e} ; so e_i is at once incremented again, as if all the intermediate vectors had been generated and discarded. Should i actually reach 1, $z = z_1$ can be output.*

Notice that the output set $\{z\}$ is probably almost sorted as it stands. We might hope that the time requirement is now also reduced to around $O(\gamma^n)$, but in practice something more like $O(\gamma^{2n})$ is observed.

By a happy accident, combined with the judicious placing of $-\infty$ and $+\infty$ sentinels at the beginning and end of $\{t\}$, the effect of a whole interval of target values can be obtained simply by presenting its endpoints in reverse (descending) order as elements of the sorted target list $\{t\}$. Finally, applying this impasse-detection to the earlier algorithm, we have

(4.5) **ALGORITHM.** *For the first stage of (4.3) use (4.4) with*

$$\begin{aligned} \mathbf{w} &= (u_{[\frac{1}{2}n]+1}, \dots, u_n), \\ \{t\} &= (-\infty, x_{\max}, x_{\min}, +\infty), \\ \{z\} &= \{y\} \text{ on output;} \end{aligned}$$

and for the second stage with

$$\begin{aligned} \mathbf{w} &= (u_0, \dots, u_{[\frac{1}{2}n]}), \\ \{t\} &= \{-\infty\} \cup \{y\} \cup \{+\infty\} \text{ sorted,} \\ \{z\} &= \{x\} \subseteq \{y\} \text{ on output.} \end{aligned}$$

As before, when $\{x\}$ is empty, \mathbf{u} is SSD0; notice that this in turn implies the analogous result for all smaller n .

Using these methods we have verified computationally the following

(4.6) **THEOREM.** \mathbf{u} is SSD0, and hence \mathbf{p} of Section 3 is SSD, for all $n \leq 79$.

This extends the calculation for $n \leq 40$ reported, but not described, in [2]. The programming language used was ALGOL68, the computer the SWURCC ICL 2980 with quadruple-precision (128-bit) REAL arithmetic, and the time approx. 6000 secs.

By (2.6) it is unnecessary to examine representations of size $2k$ in which the largest subscript occurring exceeds $T_k + 1$. This observation is of no practical value in deciding SSD0-hood for given n , for which relatively few representations are that short; however, we can modify (4.5) to restrict the signature l of representations generated to $l \leq m$ and choose $n = T_m + 1$, where m is now given. This economizes

on time and storage (now roughly $O(1.65^m)$?), and we have further established (using $m = 13$, $n = 92$) the following

(4.7) **THEOREM.** *There are no representations of zero by \mathbf{u} with signature zero and size $2k$ for $k \leq 13$ and any n whatsoever.*

[The original divide-and-conquer algorithm (4.2) has by (4.5) developed a marked asymmetry in its treatment of the two halves of \mathbf{u} , suggesting that some division ratio other than half-and-half might be more efficient; but this seems not to be the case. The c and d functions utilized in (4.4) could be computed in-line, by constantly updating current values; it is however simpler to precompute them into a triangular $2n \times n$ matrix. This initialization dominates the time for $n \leq 50$.]

5. Numerical Search for Optimal Solutions. A simple-minded exhaustive-search algorithm to find the best \mathbf{p} for given n (i.e., those with smallest p_n) might be:

(5.1) **ALGORITHM.** *Backtrack over all increasing natural vectors \mathbf{p} for which $p_n \leq u_n$ as defined by (1.12), subjecting each to a verification algorithm such as (4.1), or perhaps (4.2) if n is large enough. Few will in practice pass, and these can be output for inspection.*

This crude approach may immediately be improved using impasse-detection, since there is no point in choosing a value for p_i which is representable (see (1.2)) by previously chosen, more significant elements:

(5.2) **ALGORITHM.** *At each level i of the backtrack, compute a flag vector \mathbf{f}^i such that $f_x^i = 0$ when x is representable by (p_{i+1}, \dots, p_n) . Permissible values x for p_i correspond to flags $f_x = 1$ with $x < p_{i+1}$. Flag vectors are generated recursively using logical shift and conjunction operations, as follows:*

$$\mathbf{f}^i \leftarrow \mathbf{f}^{i+1} \& (\mathbf{f}^{i+1} \uparrow p_i) \& (\mathbf{f}^{i+1} \downarrow p_i).$$

Should the bottom level $i = 1$ be reached, a good SSD set \mathbf{p} has been constructed and is output.

The program speed is enhanced if these are programmed to take advantage of the computer's built-in bit-parallel logical operators, e.g., the 36-bit word BITS type of ALGOL 68 on the Honeywell.

An equally important but less obvious improvement involves the *lengths* of these flag vectors, that is, the h such that f_x^i is defined for all $|x| \leq h$. A naive implementation would use

$$h = \sum_{j=i+1}^n p_j,$$

which has the severe disadvantage that for small i —where most of the computation takes place—the length is at its largest. Instead, we set

$$h = \sum_{j=1}^i s_j^{i+1}$$

where s_j^i denotes the j th largest value available for p_i , calculated from \mathbf{f}^i as above; should there be less than i such values available, we have reached an impasse and must back up (increase i). With this definition, h decreases with i , while still maintaining flag vectors long enough to indicate available values of all subsequent p_j , $j < i$.

The final touch concerns rapid detection of 1's in the flag vector when searching for available x . Serial element-by-element search would be inefficient: For $n = 8$ the average number of x (5.3) is about $1\frac{1}{2}$ per 36-bit word. We utilize divide-and-conquer once more. Each nonzero word of the flag vector in turn is split in half, and the highest nonzero half is again split in half, and so on until a small enough subword length is reached to make practicable a direct table look-up for the position of the highest 1. This is deleted and the whole word scanned again for the next 1. The present implementation splits twice to lookup on 9 bits out of 36.

The overall effect on speed of these modifications increases with n , and is about 5:1 for $n = 8$. The computation eventually took about 20 hours on the UCC Honeywell level-16, producing the following tuning statistics:

$$\begin{aligned}
 &\text{size of set } n = 8 \\
 &\text{average no. of } x \text{ available} = 1.49 \\
 (5.3) \quad &\text{average available value } x = 18.7 \\
 &\text{average flag half-length } h = 31.1 \\
 &\text{average backtrack level } i = 5.93
 \end{aligned}$$

The result is that the Conway-Guy set (1.4), (1.12) is optimal for $n \leq 8$. It is not, however, always unique. For $n = 3, 5, 8$ (and $n = T_{m-1} + 2$ in general?) the triad $\{2v, 3v, 4v\}$ occurs amongst the elements of this set, where $v = \frac{1}{2}(u_n - u_{n-1})$. The $3v$ may be replaced by v while preserving SSD-hood: e.g., for $n = 3$ we have $\{2, 3, 4\}$ and $\{1, 2, 4\}$. Apart from these, the only other optimal set found is for $n = 8$:

$$(5.4) \quad \mathbf{p} = \{39, 59, 70, 77, 78, 79, 81, 84\},$$

a curiosity to which we shall return in Section 7. For $n = 9$ the computation would take 18 months; clearly, some other approach is called for.

[One striking curiosity observed is that, starting from $p_n = u_n$, the backtracker finds the Conway-Guy set at once. This is explained by Theorem (3.11), which implies that the impasse-avoidance mechanism (4.2) generates it by choosing the first available x at each level i .]

6. The Decoding Problem. In the situation where we originally encountered this subject [3, no. 5], the various subsets S of an n -set were to be uniquely encoded by the sum $x = \sum_{i \in S} p_i$, for which it is requisite that \mathbf{p} be SSD. This raises the converse problem: Given an integer x and a *weight vector* \mathbf{p} , how efficiently can it be determined whether x is a sum of distinct weights of \mathbf{p} , and (if so) of which weights—that is, what is the binary selection vector \mathbf{e} in (1.3)? [The problem is known elsewhere [11] as Knapsack Decoding, one variation of which—oddly enough—makes use of an MDCF algorithm such as we apply in Section 9 for an entirely different purpose.] Here the obvious approach is to modify (4.1) or (4.2) to

test for the required x , maintaining along with each value a (packed) record of its corresponding \mathbf{e} , in time $O(2^n)$ or $O(\sqrt{3}^n)$.

However, where \mathbf{p} is based on the Conway-Guy sequence \mathbf{u} (or a generalization), we can again improve on this performance by transforming the problem to one involving \mathbf{u} rather than \mathbf{p} . Taking the size (which we do not at this stage know) to be l , for each l from 1 to n in turn we have to find a binary \mathbf{e} such that

$$y = lu_n - x = \sum_{i=0}^{n-1} e_i u_i, \quad \text{with } \sum_{i=0}^{n-1} e_i = l, \quad e_i \in \{0, 1\}.$$

The \mathbf{e} are generated by impasse-avoiding backtrack as in (4.4), except that—since \mathbf{e} is binary rather than ternary—the c and d functions are now given by

$$c_{ij} = \sum_{k=i-j+1}^i u_k, \quad d_{ij} = \sum_{k=1}^j u_k.$$

The m of (4.4) is replaced by $n + 1$, \mathbf{w} by \mathbf{u} , and the target set $\{t\}$ contains the single (finite) value y .

When choosing e_i (in the notation of (4.4)), there are three possibilities:

$$(6.1) \quad y \notin [z_{i+1} + d_{i-1,j}, z_{i+1} + c_{i-1,j}], \quad \text{when } e_i = 1;$$

$$(6.2) \quad y \notin [z_{i+1} + u_i + d_{i-1,j-1}, z_{i+1} + u_i + c_{i-1,j-1}], \quad \text{when } e_i = 0;$$

$$(6.3) \quad y \in [z_{i+1} + u_i + d_{i-1,j-1}, z_{i+1} + c_{i-1,j}],$$

$$\begin{aligned} &\text{when either } e_i = 1, e_{i-1} = \cdots = e_{i-k} = 0 \text{ or} \\ &e_i = 0, e_{i-1} = \cdots = e_{i-k} = 1, \end{aligned}$$

where $T_{k-1} < i - 1 \leq T_k$. This follows essentially from (2.5) with $n + 1$ and m replaced by i and k , and the smaller terms ignored; what we are saying here is that a 1 in position i is equivalent to 1's in the subsequent $\sqrt{2i}$ positions, and either both are possible, (6.3), or the situation is unambiguous, (6.1), (6.2).

It follows that the time taken is at worst

$$(\text{number of choices possible for } l) \times 2^{(\text{maximum number of ambiguous positions})},$$

and since T_m is just the sum of the first m positive integers, this is just $O(n2^m) = O(n2^{\sqrt{2n}})$. It seems improbable that this can be improved to time polynomial in n .

7. Generalized Conway-Guy Sequences. The fundamental idea behind the construction of \mathbf{u} (1.12) is exposed by Theorem (3.11): Given the first n elements w_0, w_1, \dots, w_{n-1} of a sequence \mathbf{w} , set w_n equal to the smallest natural number which has no representation with signature 1 by the previous elements. Initialized with $w_0 = 0$, this procedure generates a SSD0 sequence \mathbf{w} which is apparently (as we saw in Section 4) identical with \mathbf{u} . The occurrence of an optimal SSD set (5.4) unrelated to \mathbf{u} alerts us to the possibility that other initializations might also generate interesting \mathbf{w} , in particular some which may improve on (have elements eventually smaller than) \mathbf{u} :

Some of these sequences are shown in Table 1. The first is \mathbf{u} itself, established in Section 5 to be best possible for small n . \mathbf{w}^0 almost achieves glory for $n = 8$, where it produces (5.4), but then bounces back above \mathbf{u} and deteriorates. \mathbf{w}^1 actually improves on \mathbf{u} for all $n \geq 12$ —in particular, $w_{12} = 1159 < 1164 = u_{12}$ —and \mathbf{w}^2 is better still for $n \geq 13$.

[Notice that although the sequences \mathbf{w} are SSD0, the corresponding sets \mathbf{p} may exceptionally fail to be SSD. This unruly behavior occurs in \mathbf{w}^0 for $n = 4$, where $\mathbf{p} = (1, 2, 4, 7)$ with the collision $7 = 4 + 2 + 1$; but it invariably dies out for large n , once some condition strong enough to establish (2.2) asserts itself, such as (2.1).]

In nearly all these instances, as well as in many others we have observed, a most striking property is evident:

(7.1) CONJECTURE. *After an initial settling-down region, of length n_1 possibly exceeding the length n_0 of the initializing segment, there is established a recurrence of the form*

$$w_{n+1} = 2w_n - w_{n-m} \quad \text{for } n \geq n_1,$$

where $m = \lceil \frac{1}{2} + \sqrt{2(n-r)} \rceil$ and the shift constant r depends only on \mathbf{w} .

The corresponding values of r and n_1 are shown below each sequence \mathbf{w} in Table 1, together with the appropriate limit ratio α defined as in (1.10). Using the methods of Section 4, it is a straightforward matter to establish that all the \mathbf{w} tabulated are in fact SSD0 out to $n = 67$, when extended by the recurrence (7.1). Sequences comprising an initial, arbitrary SSD0 region followed by a recurrent tail of the form detailed in (7.1), we christen *Generalized Conway-Guy Sequences*, or GCGS's. Our results suggest the following extension of (1.14):

(7.2) CONJECTURE. *Every GCGS is SSD0.*

[As r increases, so do the time and space requirements of the backtracker (4.4), presumably because the resultant bunching of elements of \mathbf{w} spoils the performance of the impasse-avoidance mechanism and produces more representations in a given interval. Strikingly, for both \mathbf{u} and \mathbf{w}^0 the stack lengths are the same for each signature l , suggesting that the actual representations generated by (4.4) are eventually dependent only on n and r , but not otherwise on \mathbf{w} .]

At this point a concrete result may be extracted from the data. In Section 1 we remarked that, given any finite SSD n -set \mathbf{p} for which $p_n/2^{n-1} = \alpha$, say, an infinite sequence of longer sets—known to be SSD and with α no worse—may be constructed by iterating (1.9). Now \mathbf{w}^2 is known to give an SSD set at $n = 67$, with already $\alpha = 0.449236 < 0.470251$; our strong interpretation of Conjecture (1.15) is thereby refuted. In fact, \mathbf{w}^3 and \mathbf{w}^4 are ultimately better still, the latter having the currently smallest known α of 0.441926. The question of bounding α below, or even of showing that it must be nonzero, remains completely open.

Good sequences all seem to have positive shifts, and indeed the evidence is that the best attainable α improves with r ; however, the final sequence in Table 1 is a curious object with a large negative shift. These are rare and have no interest from the point of view of improving α ; but they are significant in a computational search, since the large associated n_1 -value makes it difficult to decide the r -value. We suppose that the $3\frac{1}{2}$ percent of cases for which our search program failed to find any shift are all of this nature.

TABLE 1

n	u	w^0	w^1	w^2	w^3	w^4	w^{-6}
0	0	0	0	0	0	0	0
1	1	3	2	4	8	16	5
2	2	5	3	5	10	20	9
3	4	6	4	6	11	21	10
4	7	7	8	8	12	22	11
5	13	14	14	16	16	24	18
6	24	25	25	27	32	32	31
7	44	45	47	49	54	64	57
8	84	84	86	92	97	107	104
9	161	162	164	168	183	193	192
10	309	310	314	320	334	364	366
11	594	595	603	613	636	664	701
12	1164	1165	1159	1177	1218	1264	1353
13	2284	2285	2271	2262	2339	2421	2649
14	4484	4486	4456	4432	4495	4649	5194
15	8807	8810	8748	8696	8807	8934	10196
16	17305	17310	17182	17072	17280	17504	20026
17	34301	34310	33761	33531	33924	34344	39686
18	68008	68025	66919	65885	66630	67424	79006
19	134582	134885	132679	130593	130921	132427	157316
20	267420	267485	263087	258924	259503	260205	313279
r	0	0	1	2	3	4	-6
n_1	1	10	7	8	9	10	19
α	.470251	.470363	.458713	.447591	.444706	.441926	.569334

The search program which found the above specimens is itself quite involved. Very briefly, the outer loop backtracks simply over all radix- b , $(n_0 - 1)$ -digit numbers, each of which is interpreted as specifying the initial segment of an SSD0 sequence: If the n th digit has value k , then w_n is the k th value larger than w_{n-1} which preserves SSD0-hood. [The 0th, n_0 th and all subsequent digits are taken to be 1.] Then for each $n \geq 0$, the value of w_{n+1} is sought using (4.4) among the three intervals $(x_i - \varepsilon, x_i + \varepsilon)$, $i = -1, 0, +1$, where $x_i = 2w_n - w_{n-m-i}$, and ε is a parameter increased (up to a limit of 128) until a value is found, signified by a gap in the values represented by the previous elements with signature $+1$. When n reaches a predetermined setting, which will be increased (up to a limit of 50) until r is found, an attempt is made to identify r and n_1 by searching for a long, consistent sequence of relations of the form (7.1). Finally, if and when this is successful, α is calculated (see Section 9). Verification of SSD0-hood out to larger n (say 67) is invoked manually only in interesting cases.

8. Equivalent Sequences. From the (extensive) output of the search program mentioned in Section 7, it is evident that large numbers of GCGS's have exactly the same limit ratio. Such sequences we christen *equivalent*.

(8.1) LEMMA. *If c_0 and c_1 are natural numbers of opposite parity, and the sequence w is a GCGS, then so is $w' = (2w + c_1) \cup c_0$, this being the new sequence defined by $v'_0 = c_0$, $v'_{n+1} = 2w_n + c_1$ for $n \geq 0$, and sorted into ascending order of magnitude.*

Proof. If some value x has a representation by w' with signature zero which does not involve c_0 , we can at once deduce a representation of $\frac{1}{2}x$ by w with signature zero; therefore $x \neq 0$. If it does involve c_0 , x is easily seen to be odd, so again $x \neq 0$. \square

(8.2) COROLLARY. If c_q, \dots, c_0 are natural numbers such that $(c_{i+1} - c_i)/2^i$ is an odd integer for all $0 \leq i < q$, and w is a GCGS with shift r and ratio α , then

$$w' = (2^q w + c_q) \cup (c_{q-1}, \dots, c_0)$$

is a GCGS with shift $r + q$ and ratio α .

Proof. Induction on (8.1), definitions in (7.1) and (1.10). \square

Plainly, (8.2) allows us to construct indefinitely many equivalent GCGS's from any given one (such as u), with arbitrarily large shifts; the converse question—whether arbitrary equivalent GCGS's must necessarily be related in this way to some common sequence—remains open. [This last would be the unique *primitive* member w of its equivalence class such that

$$\text{HCF}(\Delta w_{n-m}, \dots, \Delta w_n) = 1$$

for $n = n_1$ and m as in (7.1), which implies a similar condition for the infinite tail of w .]

To explore the converse further, we first specify a basis for *tail* sequences: These are all sequences which eventually satisfy a recurrence of the form (7.1), without necessarily being SSD0 or even having integer elements.

(8.3) Definition. Given $0 < i < k$, let u^{ki} be the tail sequence w where

$$w_{n_1-k} = \dots = w_{n_1-k+i-1} = 0, \quad w_{n_1-k+i} = \dots = w_{n_1} = 1,$$

and w_n is given by recurrence (7.1) for $r = 0$ and $n > n_1 = T_k$. Also let

$$\alpha^{ki} = \lim_{n \rightarrow \infty} u_n^{ki} / 2^{n-1},$$

the existence of which is established in Section 9.

By choosing k suitably large and considering all the sequences involved over the region initialized in (8.3), one easily sees

(8.4) LEMMA. Given any finite set of tail sequences with shift $r = 0$, there is a k such that each tail is representable uniquely as a linear combination of the u^{ki} , $i = 0, \dots, k$, for $n > n_1$. Furthermore, if the sequences are integer, then so are the coefficients of the u^{ki} ; and if nondecreasing and nonnegative, then the coefficients are nonnegative.

There follows a short table of the first few of these basis tail sequences. Notice that $u^{11} = u$, and $u^{k0} = 1$ is the constant sequence.

	n	0	1	2	3	4	5	6	7	8	9	10	11	12	α
	m		1	2	2	3	3	3	4	4	4	4	5	5	
	u^{11}	0	1	2	4	7	13	24	44						.47025057
	u^{21}		0	1	1	2	4	7	13	25	48				.56088993/4
(8.5)	u^{22}		0	0	1	2	4	8	15	29	56				.66005617/4
	u^{31}				0	1	1	1	2	4	7	13	25	49	.63034115/32
	u^{32}				0	0	1	1	2	4	8	15	29	57	.73839425/32
	u^{33}				0	0	0	1	2	4	8	16	31	61	.79332993/32

As a special case of (8.4), the earlier basis sequences themselves can be expressed as linear combinations for given k , for instance, $\mathbf{u} = \mathbf{u}^{11} = \mathbf{u}^{20} + \mathbf{u}^{21} + 2\mathbf{u}^{22}$. This leads immediately to linear relations among the basis limit ratios: e.g., for $k = 3$,

$$\begin{aligned}\alpha^{11} &= 3\alpha^{31} + 6\alpha^{32} + 11\alpha^{33}, \\ \alpha^{21} &= \alpha^{31} + 2\alpha^{32} + 3\alpha^{33}, \\ \alpha^{22} &= \alpha^{31} + 2\alpha^{32} + 4\alpha^{33}\end{aligned}$$

may also be read off directly from (8.5).

Although the \mathbf{u}^{ki} are an integer basis, and one whose properties are readily apparent, it is a little awkward having to deal with a double sequence of basis elements. We shall investigate them further with a view of showing that, if we require only a rational basis, then the \mathbf{u}^{k1} are sufficient.

(8.6) LEMMA. *If $n > T_{k-1}$ then $u_n^{ki} \geq u_{n-1}^{ki}$.*

That is, \mathbf{u}^{ki} is monotonic, just as for (2.1). Applying the recurrence to the initial conditions of (8.3), we have

(8.7) LEMMA. *For $1 \leq i \leq k$,*

$$u_n^{ki} = \begin{cases} 2^{n-T_k} & \text{for } T_k \leq n \leq T_k + i + 1, \\ 2^{n-T_k} - 2^{n-T_k-i-2} + 1 & \text{for } T_k + i + 2 \leq n \leq T_{k+1}. \end{cases}$$

Manipulating this last, for large enough n we have

(8.8) LEMMA. *For $i \geq 2$ only,*

$$\mathbf{u}^{ki} - \mathbf{u}^{k,i-1} = \mathbf{u}^{k+1,i+1} + \sum_{j=0}^{k-i-1} 2^j \mathbf{u}^{k+1,i+2+j},$$

and similarly for the corresponding α^{ki} .

In particular, using (8.6) on the right-hand side shows that for $i \geq 2$, $u_n^{ki} \geq u_n^{k,i-1}$.

Regarding (8.8) as linear equations for the $\alpha^{k+1,i}$, they can always be solved to give $\alpha^{k+1,i}$ (and $\alpha^{k+1,1} + 2\alpha^{k+1,2}$) in terms of the α^{kj} for each $i \geq 3$. Putting this result together with (8.4), we have

(8.9) THEOREM. *The set of sequences comprising \mathbf{u}^{00} with all \mathbf{u}^{k1} for $k \geq 1$ is a basis for the GCGS's over the rationals, and similarly the set of α^{k1} for $k \geq 1$ is a rational basis for the limit ratios of GCGS's.*

[It is in practice more convenient to normalize the \mathbf{u}^{ki} by shifting downwards through $T_k - 1$, so that their ratios approach unity for large k rather than zero. For \mathbf{u}^{k1} this procedure results in the sequence commencing

$$0, 1, 2, 4, 7, 13, 25, \dots, 2^{n_1-1} - 2^{n_1-4} + 1, \dots,$$

where $n_1 = k + 2$, $r = -(T_k - 1)$, and the ratio is $2^{-r}\alpha^{k1}$. Treated in this way, it appears numerically that the \mathbf{u}^{ki} give sequences which are SSD0. On the other hand, although the \mathbf{u}^{k1} give a rational basis, they do not give an integer basis in the sense of (8.4): In fact, the appropriate $(k + 1) \times (k + 1)$ determinant—essentially $|u_j^{i1}|$ with a row of 1's attached—seems to have the value -2^{k-1} .]

Returning now to the converse of (8.2), suppose we have a pair of GCGS's with equal limit ratios α . Applying (8.2) to the one with smaller shift, we can increase its shift to the same value r as the other; then shifting the origin of n to $n = r$, we are left with two tails $\mathbf{w}^1, \mathbf{w}^2$ of shift zero and ratio α . Now consider the term-by-term difference $\mathbf{w} = \mathbf{w}^1 - \mathbf{w}^2$, which is evidently also an integer tail sequence, with ratio zero; if we could be sure that it is also nondecreasing, then by (8.4) and (8.6) it would have to be the constant sequence. Alternatively, if we could prove that the α^{k_i} are linearly independent over the rationals for given k , then the fact that \mathbf{w} is integer would do the trick. As things stand, there are oscillatory zero-ratio tails, such as multiples of $\alpha^{21}\mathbf{u}^{22} - \alpha^{22}\mathbf{u}^{21}$, one of which could conceivably equal the difference \mathbf{w} . We conjecture that this situation is impossible. The question of the linear independence of the α^{k_i} over the rationals is taken up again in Section 9.

Another question raised by (8.2) concerns the precise value of the multiplier. Specifically, if we have a GCGS \mathbf{w} , a set \mathbf{c} with $|\mathbf{c}| = q$, and an s which does not divide any value x representable by \mathbf{c} , then \mathbf{c} is SSD modulo s and $\mathbf{w}' = s\mathbf{w} \cup \mathbf{c}$ will be GCGS. It seems plausible that we could choose $s < 2^q$; after all, for large enough q there are many SSD sets \mathbf{c} at our disposal, representing x much more dispersed and riddled with gaps for potential s than that used above—which is effectively $\mathbf{c} = (1, 2, \dots, 2^{q-1})$. Now we would have constructed a \mathbf{w}' with ratio $(s/2^q)\alpha$ actually smaller than α , and might continue thus to reduce it indefinitely. (Of course, $s > 2^q$ is of no interest: α will be worse, and \mathbf{w}' will lack the locally minimal property mentioned in earlier sections.)

Unhappily, we discover

(8.10) LEMMA. *With the above notation, $s \geq 2^q$.*

Proof. Otherwise, let $\beta = \sqrt[q]{s} < 2$, and construct a new sequence \mathbf{w}'' from \mathbf{c} alone via the rule

$$v_n'' = s^i c_j \quad \text{where } n = iq + j.$$

Then

$$v_n''/\beta^n < \max_j (c_j/\beta^j),$$

and so

$$v_n''/(2^n/n) \rightarrow 0 \quad \text{since } \beta < 2.$$

However, if we assume that \mathbf{w}'' is such that (2.2) can be proved for it, and associate a set \mathbf{p} with it as in (1.4), by (7.2) \mathbf{p} should be SSD. Since the 2^n subset-sums are distinct, their maximum is at least $2^n - 1$, the maximum w_n of \mathbf{p} is at least $(2^n - 1)/n$, and the above limit must be at least 1. \square

9. Computation of Limit Ratios. In order to carry out a search for linear relations between certain sets of limit ratios of GCGS's, or to examine whether a particular ratio (such as α_u) is algebraic of some given degree, it is first necessary to compute these quantities to very high accuracy. The simple-minded approach of just calculating $u_n/2^{n-1}$ for some suitably large n is, as we shall see, convergent with order only $\frac{1}{2}$; at the cost of some effort, this dismal performance can be improved.

Suppose then that w is a GCGS with shift r and ratio α , and write $a_n = w_n/2^{n-1}$. Let m and l be defined by

$$n - r = T_{m-1} + l, \quad \text{where } 1 \leq l \leq m.$$

The fundamental recursion (7.1) may be rewritten in the form

$$a_{n+1} = a_n - a_{n-m}/2^{m+1},$$

from which it may immediately be inferred that $\alpha = \lim a_n$ exists, since a_n is decreasing and positive. Iterating this recursion, for large j we find

$$a_{n+j} = a_n - (a_{n-m} + \cdots + a_{T_{m-1}+r})/2^{m+1} - (a_{T_{m-1}+r} + \cdots + a_{T_m+r})/2^{m+2} - \cdots$$

Letting $j \rightarrow \infty$, this is easily rearranged to give

$$(9.1) \quad a_n = \alpha + \sum_{t=m}^{\infty} 2^{-t-1} \sum_{i=0}^{t-1} a_{T_{t-1}+r+i} - 2^{-m-1} \sum_{i=0}^{l-2} a_{T_{m-1}+r+i}.$$

Finally, the right-hand side of (9.1) may be iteratively substituted for the a 's occurring within it, giving an expansion which will turn out to be of the general form

$$(9.2) \quad a_n/\alpha = \sum_{k=0}^{\infty} U_k(l, m),$$

where for fixed l the U_k are of the form (polynomial in m)/(2^k)^m. This expansion is asymptotic rather than convergent, converging only while $k < m$ and afterwards diverging. Its first few terms are

$$\begin{aligned} a_n/\alpha = 1 + \left[m - \frac{1}{2}l - \frac{3}{2} \right] 2^{-m} \\ + \left[\left(m - \frac{1}{2}l \right)^2 + \frac{8}{3}m - \frac{5}{4}l + \frac{37}{18} - \frac{1}{2}\delta_{1l} \right] 4^{-m} + \cdots, \end{aligned}$$

where $\delta_{ij} = 1$ if $i = j$, 0 otherwise.

To find a recurrence for the U_k , the expansion (9.2) must be substituted into (9.1). This is complicated by the fact that, in the innermost summation over i in (9.1), the first term has an m -value of $t-2$, whereas all the rest have $t-1$. It is therefore convenient to define U_k also for $l=0$, resulting in the following "unified" recurrence:

$$(9.3) \quad U_k(l, m) = \begin{cases} 1 & \text{if } k = 0, \\ U_k(m-1, m-1) & \text{if } l = 0, \\ \sum_{t=m}^{\infty} 2^{-t-1} \sum_{i=0}^{t-1} U_{k-1}(i, t-1) \\ \quad - 2^{-m-1} \sum_{i=0}^{l-2} U_{k-1}(i, m-1) & \text{if } k, l > 0. \end{cases}$$

This form is not very convenient computationally, since—on account of the complication mentioned above— U_k is not polynomial in l for $l < k$. We therefore split it apart notationally into $U'_k(l, m)$ for $l \geq k$, and $U''_k(l, m)$ for $l \leq k$, where now the functional parentheses indicate *polynomial* dependence (apart from an easily accommodated factor of 2^{-km}). Introducing the intermediate polynomials $V_k(m)$, $S'_k(l, m)$,

$S''_{kl}(m)$ for convenience, we write down the “polynomial” recurrence for $U'_k(l, m)$, $U''_{kl}(m)$:

$$\begin{aligned}
 V_k(m) &= \begin{cases} 1 & \text{if } k = 0, \\ \sum_{t=m}^{\infty} [S''_{kk}(t) + S'_k(t+1, t)] & \text{if } k > 0; \end{cases} \\
 S'_k(l, m) &= \begin{cases} 0 & \text{if } l \geq k = 0, \\ 2^{-m-1} \sum_{i=k-1}^{l-2} U'_{k-1}(i, m-1) & \text{if } l \geq k > 0; \end{cases} \\
 S''_{kl}(m) &= \begin{cases} 0 & \text{if } k \geq l = 0, \\ 2^{-m-1} \sum_{i=0}^{l-2} U''_{k-1,i}(m-1) & \text{if } k \geq l > 0; \end{cases} \\
 U'_k(l, m) &= \begin{cases} V_k(m) - S''_{kk}(m) - S'_k(l, m) & \text{if } l \geq k \geq 0; \end{cases} \\
 U''_{kl}(m) &= \begin{cases} U'_k(m-1, m-1) & \text{if } k \geq l = 0, \\ V_k(m) - S''_{kl}(m) & \text{if } k \geq l > 0. \end{cases}
 \end{aligned}
 \tag{9.4}$$

In this form the recurrence is suitable for programming on a symbolic algebra package, such as MACSYMA [7] or REDUCE [8]. Setting $l = 1$ in (9.2) and noticing that for $k > 0$, $U''_{k1}(m) = V_k(m) = W_k(m)2^{-km}$, say, we find

$$a_n/\alpha = \sum_{k=0}^{\infty} W_k(m)2^{-km} \quad \text{where } n = T_{m-1} + 1,$$

and MACSYMA's formal summation and limit facilities may (if MACSYMA is in a good mood) deliver the following values for $W_k(m)$ in a few minutes:

$$\begin{aligned}
 W_0 &= 1, \\
 W_1 &= m + 1, \\
 W_2 &= m^2 + \frac{5}{3}m + \frac{14}{9}, \\
 W_3 &= \frac{4}{3}m^3 + \frac{8}{3}m^2 + \frac{44}{9}m + \frac{32}{9}, \\
 W_4 &= \frac{8}{3}m^4 + \frac{16}{3}m^3 + \frac{136}{9}m^2 + \frac{2768}{135}m + \frac{18368}{2025}, \\
 W_5 &= \frac{128}{15}m^5 + \frac{128}{9}m^4 + \frac{1664}{27}m^3 + \frac{15488}{135}m^2 + \frac{151808}{2025}m + \frac{32768}{2025}, \\
 W_6 &= \frac{2048}{45}m^6 + \frac{2048}{45}m^5 + \frac{10240}{27}m^4 + \frac{333824}{405}m^3 + \frac{90112}{225}m^2 \\
 &\quad - \frac{1073152}{8505}m - \frac{220921856}{2679075}.
 \end{aligned}
 \tag{9.5}$$

The general form of these polynomials is not immediately obvious, although the coefficients have tantalizingly small prime factorizations. However, the leading term is accessible as a special case of the observation that

$$U_k(l, m) = (m - \tfrac{1}{2}l)^k 2^{T_{k-1}-km}/k! + (\text{terms in } m, l \text{ jointly of degree } < k),$$

proved easily by substituting the right-hand side into the recurrence (9.3). For any fixed l , therefore, including $l = 1$, the error involved in terminating the expansions (9.2) or (9.5) at the $(k - 1)$ th term is essentially

$$(9.7) \quad m^k 2^{k(k-1)/2 - km} / k!.$$

For example, with $k = 10$, $m = 26$, $n = 352$, and $\mathbf{w} = \mathbf{u}$ the original Conway-Guy sequence, we find that the actual error in (9.5) and the estimate of (9.7) are 10^{-61} times 0.20 and 0.27, respectively.

The special case of (9.7) with $k = 1$ shows that the error in taking a_n itself as an approximation to α is about $m/2^m$, or roughly $2^{-\sqrt{m}}$: That is, the obvious method converges with order $\frac{1}{2}$. Setting $k = m$ in (9.7), we see that the error becomes (roughly) $2^{-\frac{1}{2}m^2} m^m / m!$, or (even more roughly) 2^{-n} , so that the convergence of this method is first-order. It does require us to compute the polynomials (9.6) initially; on the other hand, this is done once-and-for-all and may then be used to compute the limit ratio for many different sequences.

A less complex method involves ignoring the details of the U_k 's using instead only knowledge of their general form, and employing a modified Richardson extrapolation procedure (as used in Romberg integration; see [6, Chapters 14 and 17]) for accelerating the convergence of a_n to α . Define

$$(9.8) \quad \begin{aligned} b_{0m} &= a_{(T_m+1+r)}, \\ b_{k+1,m} &= \Delta_i^{k+2} (2^i b_{k,m+i}) / (2^{k+1} - 1)^{k+2}, \end{aligned}$$

where Δ_i^k denotes k th order differencing with respect to i (rather than m). The algorithm proceeds by increasing m until a pair of approximations is encountered whose difference is less than the required accuracy. It is only necessary to store an advancing "diagonal" of T_{k+1} values, comprising the current b_{km} and k intermediate differences for each.

We briefly summarize the error analysis. Suppose that the calculation has proceeded as far as b_{0m} for $m = m_0$, and we wish to find the $k = k_1$ for which the error in $b_{k-1,m}$ is minimized. Referring to (9.2), all U_k for $k < k_1$ will have been eliminated by the algorithm, and the leading term of the residual error will be of the form (9.7) as transformed by the algorithm. Referring to (9.8), the effect of each stage k is crudely to divide the residual error by $(2^k - 1)^k$, or roughly 2^{k^2} ; so by stage k it will have been reduced by a factor of $2^{\sum k^2}$, or roughly $2^{\frac{1}{3}k^3}$. Ignoring all but the factor 2^{-km} of (9.7), we need to minimize

$$2^{-\frac{1}{3}k^3 - km}$$

subject to the constraint arising from the way m is reduced by $k + 1$ in going from stage $k - 1$ to k : That is, $m + \sum k = m_0$, or roughly

$$m + \frac{1}{2}k^2 = m_0.$$

Solving this simple calculus problem gives the minimum at $k_1 = \sqrt{\frac{1}{2}m_0}$, $m_1 = \frac{3}{4}m_0$. The error at this point is roughly $2^{-(11/12)m_0^{3/2}}$, or $c^{n^{3/4}}$ for suitable constant c ; so this method is of order $\frac{3}{4}$. For 60 significant figures and $\mathbf{w} = \mathbf{u}$ as before, we need $m = 45$, $n = 1036$, $k = 8$.

TABLE 2

k	$\alpha^{k1} \times 2^{T_k-1}$
(v)	0.633368347305411640436713144616576659293359908783416689204401
(u)	0.470250569622349731271176348783758019761969039167059197165356
2	0.560889931839105275636384948015840006734260398171595554263428
3	0.630341152953289800442032143698140347343856000669900313548808
4	0.677669268053777242613587345333872800584971863152808709070330
5	0.707664435666270871027823533000164520492533323753898941029982
6	0.725790229709249261978251175393256303351188007190298345209967
7	0.736392117117017642888896676688968317465731121473706835171727
8	0.742451223289530587380718963519922408252517958351565949689771

For α_v , $\alpha_u = \alpha^{11}$, and $\alpha^{21}, \dots, \alpha^{81}$ as defined in Section 8 and suitably normalized, we find the 60-place values shown in Table 2, using Richardson extrapolation and MACSYMA's *bigfloat* arithmetic.

With these, we conducted a Multi-Dimensional Continued Fraction (MDCF) search for an integer relation satisfied by the first d of the α^{k1} , $k = 1, \dots, d$, and in addition for some integer polynomial of degree $d - 1$ satisfied by α_u (1.12), and by α_v (1.6). There follow in (9.9) the lower bounds obtained for the height of such an object, that is, the maximum absolute value of its coefficients. For example, the entries under $d = 2$ imply that no equation of the form $a\alpha^{11} + b\alpha^{21} = 0$ holds with height less than 30 digits, nor similarly $a + b\alpha_u = 0$ nor $a + b\alpha_v = 0$. The method used to compute these relation height bounds we have christened the *Parallelotope* algorithm; we hope to describe it elsewhere [10]. The general topic of MDCF algorithms is discussed in [9], and a detailed description of the recent " L^3 " algorithm is presented in [5, Section 1]. For the moment, we content ourselves with remarking that, for given accuracy δ ($= \frac{1}{2}10^{-60}$ here), the maximum theoretically obtainable bound is of order $\delta^{-1/d}$.

d	2	3	4	5	6	7	8
α^{k1}	5.3×10^{29}	3.3×10^{19}	2.8×10^{14}	1.8×10^{11}	5.9×10^8	6.8×10^6	8.9×10^3
(9.9) α_u	1.7×10^{30}	7.0×10^{19}	9.2×10^{13}	7.5×10^9	4.5×10^9	2.8×10^5	1.4×10^6
α_v	4.8×10^{29}	4.5×10^{19}	5.0×10^{14}	1.6×10^{11}	9.6×10^8	1.7×10^6	2.0×10^5
max	1.4×10^{30}	1.3×10^{20}	1.2×10^{15}	1.1×10^{12}	1.1×10^{10}	4.1×10^8	3.4×10^7

Department of Computing Mathematics
University College, Cardiff
Senghennydd Road
Cardiff CF2 4AG, England

1. R. K. GUY, "Sets of integers whose subsets have distinct sums," *Ann. Discrete Math.*, v. 12, 1982, pp. 141–154.
2. R. K. GUY, *Unsolved Problems in Number Theory*, Springer-Verlag, Berlin and New York, 1981.
3. M. GARDNER, *Science Fiction Puzzle Tales*, Penguin, 1981.
4. R. SEDGEWICK, *Algorithms*, Addison-Wesley, Reading, Mass., 1983.
5. A. K. LENSTRA, H. W. LENSTRA & L. LOVÁSZ, "Factoring polynomials with rational coefficients," *Math. Ann.*, v. 261, 1982, pp. 515–534.
6. C.-E. FRÖBERG, *Numerical Mathematics*, Benjamin Cummings, Menlo Park, Calif., 1985.

7. *MACSYMA Reference Manual*, MIT, Cambridge, 1982.
8. A. C. HEARN (editor), *REDUCE User's Manual* 3.1, Rand Corporation, Santa Monica, Calif., 1984.
9. A. J. BRENTJES, *Multi-Dimensional Continued Fraction Algorithms*, Math. Centre Tracts 145, Mathematisch Centrum, Amsterdam, 1981.
10. W. F. LUNNON, *MDCF Algorithms and their Applications*, Proc. Cardiff Conference on the Application of Computers to Mathematics, University College, Cardiff, 1986.
11. J. C. LAGARIAS & A. M. ODLYZKO, "Solving low-density subset problems," *J. Assoc. Comput. Mach.*, v. 32, 1985, pp. 229–246.
12. M. D. ATKINSON, A. NEGRO & N. SANTORO, *Integer Sets with Lexicographically Ordered Subset Sums*, Technical Report no. 100, Carlton Univ., 1986.