MATHEMATICS OF COMPUTATION
VOLUME 36, NUMBER 154
APRIL 1981

Algorithms for Computing the h-Range of the Postage Stamp Problem

By Svein Mossige

Abstract. New algorithms, based on a very efficient method to compute the h-range, have been used to extend known tables of the extremal h-range, to complete the solution in the case $k = 3$, and to find a lower bound for the extremal 2-range.

1. Introduction. The postage stamp problem consists of choosing, for given h and k, a set of k positive integers such that

(a) sums of h (or fewer) of these integers can realize the numbers $1, 2, \ldots, n$;

(b) the value of n in (a) is as large as possible.

Let h and k be given positive integers and $A_k = \{a_1, a_2, \ldots, a_k\}$ a set of distinct integers such that $1 = a_1 < a_2 < \cdots < a_k$. We form the set of linear combinations

$$S = \left\{ \sum_{i=1}^{k} x_i a_i;\ x_i \geqslant 0,\ \sum_{i=1}^{k} x_i \leqslant h \right\}.$$

The number n is called the *h-range* of the *basis* A_k if the set S contains the integers $1, 2, \ldots, n$, that is $[1, n] \subseteq S$, and $n + 1 \notin S$. The h-range of A_k is denoted by $n_h(A_k)$.

For given h and k, an *extremal basis* is a basis of k elements for which n is as large as possible. The corresponding *extremal h-range* is denoted by $n_h(k)$.

The set A_k is called an *admissible* basis if

(1) $\qquad a_i < a_{i+1} < n_h(A_i) + 2, \qquad i = 1, 2, \ldots, k - 1$, that is, $n_h(A_k) \geqslant a_k$.

In what follows, only such bases A_k will be considered. Let h_0 denote the smallest possible h such that A_k is an admissible basis.

For all k and $h \geqslant h_0$, we trivially have

$$n_{h+1}(A_k) \geqslant n_h(A_k) + a_k. \tag{2}$$

Further, Selmer [10] proved that, for arbitrary k and $h \geqslant h_0$,

$$n_h(A_k) \geqslant (h + 1)a_{k-1} - a_k \tag{3}$$

implies

$$n_{h+1}(A_k) = n_h(A_k) + a_k. \tag{4}$$

If h is increased by 1, the right-hand side of (3) increases with a_{k-1}, while the left-hand side increases with at least a_k. There is consequently an h_1 ($\geqslant h_0$) such that (3) and hence (4) are satisfied for all $h \geqslant h_1$. This means that for given h,

Received October 10, 1979; revised July 10, 1980.
1980 *Mathematics Subject Classification.* Primary 10J79, 10–04.

$h \geqslant h_1$, we have

$$n_h(A_k) = n_{h_1}(A_k) + (h - h_1)a_k. \tag{5}$$

2. Recursive Definition of the Set S. To simplify the notation, we introduce the artificial basis element $a_0 = 0$. For given h, k, and A_k, the set $S = s(h, k)$ can then be defined recursively by

$$\begin{cases} s(1, k) = A_k \cup \{0\}, \\ s(r, k) = \bigcup_{i=0}^{k} (a_i + s(r - 1, k)), \qquad r = 2, 3, \ldots, h, \end{cases} \tag{6}$$

where, as usual,

$$a_i + s(r - 1, k) = \{a_i + b; b \in s(r - 1, k)\}. \tag{7}$$

We further need the obvious relation

$$s(h, k) = \bigcup_{i=0}^{h} (ia_k + s(h - i, k - 1)). \tag{8}$$

The argument $k - 1$ indicates that a_k is removed from the basis. We define $s(0, k) = s(h, 0) = \{0\}$.

From (6) and (7), we see that $s(r, k)$ is a union of the set $s(r - 1, k)$ and the sets given by adding a_i ($i \geqslant 1$) to each element in $s(r - 1, k)$. The following interpretation of the process may be enlightening:

Let $b_0 = 0, b_1, \ldots, b_v$, be all the elements in the set $s(r - 1, k)$, and mark a ruler at distances $b_0, b_1, \ldots, b_v$ from its starting point. Place this point at the origin of an axis and transfer to the axis the marks on the ruler. Then translate the ruler along the axis, first a_1 units and transfer the marks to the axis, then a_2 units, etc., up to a_k units and transfer each time the marks on the ruler to the axis. The numbers corresponding to the marks on the axis are then exactly the elements of the set $s(r, k)$.

(8) suggests a method to compute $n_h(A_{k-1} \cup \{a_k\})$ based on the information achieved by the computation of $n_h(A_{k-1})$.

3. The h-Range (Basic Algorithm). In the computer, we represent the set $s(r, k) \setminus \{0\}$, $1 \leqslant r \leqslant h$, as a bit string B_r of length ra_k:

$$B_r\text{: bit } t = 1 \quad \text{iff } t \in s(r, k) \setminus \{0\}.$$

In particular, B_1 corresponds to the basis A_k.

Let a_iB_{r-1} denote the bit string B_{r-1} *shifted* a_i places to the right (filling in with zeros to the left). Then the recursive definition (6) is equivalent to the following set of OR operations:

$$B_r = \bigvee_{i=0}^{k} a_iB_{r-1}, \qquad r = 2, 3, \ldots, h. \tag{9}$$

Each time a new B_r is constructed, we check if the a_k left-most bits in B_r are 1 bits. When this is the case for the first time, we have $n_r(A_k) \geqslant a_k$ and hence $h_0 = r$. For each $r \geqslant h_0$, $n_r(A_k)$ is determined by scanning the bit string B_r. The first zero is then in position $n_r(A_k) + 1$.

If $h_0 < h$, we check for each r, $h_0 \leqslant r < h$, whether (3) is satisfied. If this occurs for the first time for $r = h_1$, we use (5) to determine $n_h(A_k)$. For large h, this simple device yields a significant reduction of the computing time.

Another device, also very effective for large h, is to reduce the number of registers needed to hold the bit strings B_r. For $r \geqslant h_0$, we can delete the first $n_r(A_k)$ 1 bits. From (2), it follows that no bit string length for $r \geqslant h_0$ will then exceed $h_0 a_k - n_{h_0}(A_k)$.

When computing $n_h(k)$ by the method of Section 4, we may also save a substantial amount of time by chopping off the bit strings B_r at the other end. Keeping k fixed, the extremal range $n_h(k)$ turns out to increase fairly regularly with h. We can therefore estimate a safe upper bound for $n_h(k)$ and delete those parts of the bit strings B_r which exceed this bound.

4. The Extremal h-Range $n_h(k)$, $h > 2$. For given h and k, let U_h be the universe of all admissible sets A_k, defined by the conditions (1). To find the set(s) with the extremal h-range, we scan the universe U_h. For each $A_k \in U_h$, the h-range $n_h(A_k)$ is computed by the basic algorithm.

An alternative approach (not used in our calculations) is to extract from the basic algorithm also the ranges $n_r(A_k)$ for $r < h$. This would allow for a simultaneous computation of extremal ranges for all $r \leqslant h$, at the cost of larger storage requirements.

If we know a lower bound L such that $n_h(k) \geqslant L$, we can skip all $A_k \in U_h$ with $ha_k < L$. Having already calculated $n_{h-1}(k)$ and a corresponding extremal basis A_k^*, we may start the scanning of U_h with $L = n_{h-1}(k) + a_k^*$. Whenever we find a larger $n_h(A_k)$, this can replace the previous bound L. To get large bounds as quickly as possible, we scan the intervals (1) for a_i downwards. This also simplifies the exclusion of all A_k with $ha_k < L$.

5. The Extremal 2-Range $n_2(k)$. For $h = 2$, the devices mentioned at the end of Section 3 are of no use. We may instead utilize (8), which now takes the form

$$s(2, k) = s(2, k-1) \cup (a_k + (A_{k-1} \cup \{0\})) \cup \{2a_k\}. \tag{10}$$

We scan U_2 for $k - 1$, corresponding to $s(2, k-1)$. Each A_{k-1} is then extended to A_k, where $a_{k-1} < a_k < n_2(A_{k-1}) + 2$ by (1). In the computer representation of (10) as a bit string, we may clearly delete $\{a_k\}$ (since A_k is admissible). We may also delete $\{2a_k\}$, if we note that a calculated 2-range $2a_k - 1$ then really corresponds to the maximal range $2a_k$. As a result, we get the extremal ranges $n_2(k-1)$ and $n_2(k)$ simultaneously.

Because of the simplicity of (10), this method leads to a surprisingly large saving in computing time. A simultaneous computation of $n_2(9)$ and $n_2(10)$ used only 25% of the time needed to determine these extremal ranges individually by the method of Section 4.

Also for small $h > 2$, the above method may represent an interesting alternative to such individual computations.

Riddell and Chan [7] have another algorithm for $h = 2$. Their method can be used for $h > 2$, but then their universe of sets is larger than necessary, including also nonadmissible sets.

Lunnon's algorithm [4] utilizes (8) for *all h*. For fixed k, this requires the storing of all bit strings $s(h - i, k - 1)$, $i = 0, 1, \ldots, h$.

6. The Extremal 2-Range for Symmetric Bases. A set $A_k \cup \{a_0 = 0\}$ is *symmetric* if $a_i + a_{k-i} = a_k$ for $i = 0, 1, \ldots, k$. The 2-range of a symmetric basis is $2a_k$; cf. [8].

To compute the extremal 2-range for symmetric bases, it is sufficient to scan all admissible sets $A_{k/2}$ for k even and all sets $A_{(k+1)/2}$ for k odd. The computing time can be reduced by the methods described at the end of Section 4. The amount of work is comparable to that of computing $n_2([(k + 1)/2])$.

7. Results of the Calculations. The algorithms were written in assembler language and Fortran and performed on the Univac computer at the University of Bergen.

For $k = 2$, explicit formulas for the extremal bases and ranges are well known; cf. [11].

In 1968, Hofmeister [2] solved the corresponding problem for $k = 3$ almost completely, giving formulas which are valid for *sufficiently large h*. He also gave a table for $h \leqslant 34$ (where the "anticipated" extremal basis $\{1, 19, 102\}$ for $h = 22$ is missing). Recently, Hofmeister [3] has shown that it suffices to check separately the cases with $h \leqslant 200$. Using results from Selmer's paper [10], we have performed this check. It turns out that for $h > 22$, *Hofmeister's formulas for* $k = 3$ *cover all extremal bases.*

For $k \geqslant 4$, the standard table of reference is that of Lunnon [4]. In addition, Seldon [9] computed $n_3(10)$, Phillips [6] $n_4(8)$, and Riddell and Chan [7] $n_2(13)$.

Our programs used only 4.5 hours to verify Lunnon's tables. This figure reflects the speed of Univac 1100/82, but even more the efficiency of our basic bit string algorithm.

We have extended earlier tables in four directions:

For $k = 4$, Table 1 below lists $n_h(4)$ and the corresponding extremal bases for $2 \leqslant h \leqslant 28$.

TABLE 1

Extremal h-ranges for $k = 4$

h	$n_h(4)$	a_2	a_3	a_4	h	$n_h(4)$	a_2	a_3	a_4
2	12	3	5	6	15	1383	12	65	240
3	24	4	7	8	16	1650	11	78	216
4	44	3	11	18	17	1935	11	90	252
5	71	4	12	21	18	2304	16	73	338
		5	12	28	19	2782	10	99	360
6	114	4	19	33	20	3324	16	103	488
7	165	5	24	37	21	3812	16	103	488
8	234	6	25	65	22	4368	12	121	561
9	326	5	34	60	23	5130	14	142	659
10	427	6	41	67	24	5892	16	163	757
11	547	7	48	85	25	6745	20	149	860
12	708	7	48	126	26	7880	16	194	734
13	873	9	56	155	27	8913	21	177	1006
14	1094	8	61	164	28	9919	21	177	1006

Table 2 gives some extremal ranges for $k = 5$ and $k = 6$, with bases.

TABLE 2
Some $n_h(k)$ for $k > 4$

k	h	$n_h(k)$	a_2	a_3	a_4	a_4	a_6
	10	1055	8	27	119	194	
5	11	1475	10	34	165	270	
	12	2047	10	26	195	320	
6	7	664	7	12	64	113	193
	8	1045	9	14	65	170	297

Table 3 shows the three extremal bases corresponding to $n_2(14) = 80$. The first one of these bases inspired the appendix below.*

TABLE 3
Extremal bases for $n_2(14) = 80$

a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}
2	5	8	11	14	17	20	23	24	25	51	53	55
3	4	5	8	14	20	26	32	35	36	37	39	40
3	4	9	10	15	16	21	22	24	25	51	53	55

Table 4 gives the extremal 2-ranges for symmetric bases with $15 \leqslant k \leqslant 30$. The set of differences $d_i = a_i - a_{i-1}$ is also symmetric, and is only listed up to the middle of the set.

TABLE 4
The extremal symmetric bases, $h = 2$, $15 \leqslant k \leqslant 30$

k	$n_2(A_k)$	$d_i = a_i - a_{i-1}, i = 1, 2, \ldots, \left[\frac{k+1}{2}\right]$
15	92	1 2 1 1 3 6 6 6
16	104	1 2 1 1 3 6 6 6
17	116	1 2 1 1 3 6 6 6 6
18	128	1 2 1 1 3 6 6 6 6
19	140	1 2 1 1 3 6 6 6 6 6
20	152	1 2 1 1 3 6 6 6 6 6
21	164	1 2 1 1 3 6 6 6 6 6 6
	164	1 2 1 2 4 3 2 6 8 8 8
22	180	1 2 1 2 4 3 2 6 8 8 8
23	196	1 2 1 2 4 3 2 6 8 8 8 8
24	212	1 2 1 2 4 3 2 6 8 8 8 8
25	228	1 2 1 2 4 3 2 6 8 8 8 8 8
26	244	1 2 1 2 4 3 2 6 8 8 8 8 8
	244	1 2 1 1 3 3 4 1 9 9 9 9 9
27	262	1 2 1 1 3 3 4 1 9 9 9 9 9 9
28	280	1 2 1 1 3 3 4 1 9 9 9 9 9 9
29	298	1 2 1 1 3 3 4 1 9 9 9 9 9 9 9
30	316	1 2 1 3 1 1 7 1 4 3 11 11 11 11 11
	316	1 2 1 2 2 4 5 2 6 4 10 10 10 10 10
	316	1 2 1 1 3 3 4 1 9 9 9 9 9 9 9
	316	1 1 3 1 2 5 1 3 2 10 10 10 10 10 10

*$n_2(15) = 92$ and $n_2(16) = 104$ with the extremal bases given in Table 4.

Appendix, by Torleiv Kløve and Svein Mossige. Rohrback [8] proved that

$$n_2(k) \geqslant \frac{1}{4}k^2 + O(k)$$

and conjectured that this is best possible. However, Hämmerer and Hofmeister [1] proved that

$$n_2(k) \geqslant \frac{5}{18}k^2 + O(k).$$

We shall prove that

$$n_2(k) \geqslant \frac{2}{7}k^2 + O(k).$$

The same result has been proved independently by Mrose [5] by a different and more complicated construction.

We use the following notation: Let a, b, and c be positive integers such that $b \geqslant a$ and c divides $b - a$, and let d be a nonnegative integer. Then

$$\begin{aligned} [a(c)b] &= \{a + ic \mid 0 \leqslant i \leqslant (b-a)/c\}, \\ [a, b] &= [a(1)b], \\ B(a, d) &= [a(d)a + d^2]. \end{aligned}$$

LEMMA. *Let x, y, and z be positive integers where $y \geqslant x \geqslant 2$, and let*

$$\begin{aligned} S_1 &= [0, x-1], \\ S_2 &= [x - 1(x)yx - 1] \cup \{0\}, \\ S_3 &= [yx - 1, yx + x - 2] \cup \{0\}, \end{aligned}$$

and $T = S_1 \cup S_2 \cup S_3$. Then

(I) $T + T = [0, 2yx + 2x - 4]$,

(II) $T + B(z, x-1) = [z, z + yx + x^2 - x - 1]$.

Proof. The proof of (I) is divided into five parts:

$$\begin{aligned} &[0, 2x-2] && = S_1 + S_1, \\ &[2x - 1, yx + x - 2] && \subset S_1 + S_2, \\ &[yx, yx + 2x - 3] && \subset S_1 + S_3, \\ &[yx + x - 2, 2yx + x - 3] && \subset S_2 + S_3, \\ &[2yx, 2yx + 2x - 4] && \subset S_3 + S_3, \end{aligned}$$

the verification of which is straightforward. To prove (II), we first note that

$$[z, z + x(x-1)] \subset S_1 + B(z, x-1).$$

Next, if $u \in [z + x^2 - x, z + yx - 1]$, then $u = z + lx - m$, where $1 \leqslant m \leqslant x$ and $x \leqslant l \leqslant y$. Hence,

$$u = ((l - m + 1)x - 1) + (z + (m-1)(x-1)) \in S_2 + B(z, x-1),$$

and so

$$[z + x^2 - x, z + yx - 1] \subset S_2 + B(z, x-1).$$

Further,

$$[z + yx, z + yx + x^2 - x - 1] \subset S_3 + B(z, x - 1).$$

If $w \in T + B(z, x - 1)$, then

$$z = 0 + z \leqslant w \leqslant yx + x - 2 + z + (x - 1)^2 = z + yx + x^2 - x - 1,$$

and (II) follows.

THEOREM. *Let $k \geqslant 8$ be an integer, and put $x = \lceil k/7 \rceil$, $y = k - 4x + 3$. Further, let T have the same meaning as in the lemma, and let $u = n_2(T) + 1$, $v = n_2(T \cup B(u, x - 1)) + 1$ and*

$$A = (T \setminus \{0\}) \cup B(u, x - 1) \cup B(v, x - 1).$$

Then, $|A| = k$ and

$$n_2(A) = 4kx - 14x^2 + 12x - 4 = \frac{2}{7}k^2 + \frac{12}{7}k + O(1).$$

Proof. By the lemma,

$$u = 2yx + 2x - 3,$$
$$v = u + yx + x^2 - x = 3yx + x^2 + x - 3.$$

Hence,

$$T + A = [1, v + yx + x^2 - x - 1] = [1, 4yx + 2x^2 - 4].$$

Let $w = 4yx + 2x^2 - 3$. Suppose $w = a + b$ where $a, b \in A$, $a \leqslant b$. Then $w \leqslant 2b$, and so $b > 2yx + x^2 - 2$. Hence, $b \in B(v, x - 1)$. Similarly, $w \geqslant 2a$, and so $a < 2yx + x^2 - 1$. Hence, $a \notin B(v, x - 1)$. Since $w \notin T + A$, $a \in B(u, x - 1)$. Let

$$a = (2yx + 2x - 3) + l(x - 1), \qquad 0 \leqslant l \leqslant x - 1,$$
$$b = (3yx + x^2 + x - 3) + m(x - 1), \quad 0 \leqslant m \leqslant x - 1.$$

Then,

$$w = 4yx + 2x^2 - 3 = a + b = 5yx + x^2 + 3x - 6 + (l + m)(x - 1).$$

Hence, $(l + m + 3)(x - 1) + x(y - x) = 0$. However, since $k \geqslant 8$, $x - 1 > 0$ and $y - x > 0$, and so we have a contradiction. Therefore,

$$n_2(A) = 4yx + 2x^2 - 4 = 4kx - 14x^2 + 12x - 4.$$

If we put $x = k/7 + \theta(k)$, then $0 \leqslant \theta(k) < 1$, and $\theta(k)$ depends only on k modulo 7. Further,

$$n_2(A) = \frac{2}{7}k^2 + \frac{12}{7}k + (12\theta - 14\theta^2 - 4).$$

Remark. One extremal basis for $k = 14$ is of the form $T \cup B(u, x - 1)$ with $x = 3$ and $y = 8$. For general k, one optimal choice for x and y in this case is $x = \lceil 3k/16 \rceil$ and $y = k - 3x + 3$, which gives

$$n_2(T \cup B(u, x - 1)) = \frac{9}{32}k^2 + O(k).$$

This is asymptotically weaker than the construction above. A more general construction would be

$$A = (T \setminus \{0\}) \cup \bigcup_{i=1}^{t} B(u_i, x - 1),$$

where $u_i = n_2(T) + 1 + (i - 1)(yx + x^2 - x)$. The construction above corresponds to $t = 2$, which turns out to give the asymptotically best result.

Universitetet i Bergen
Matematisk Institutt
5014 Bergen, Norway

1. N. Hämmer & G. Hofmeister, "Zu einer Vermutung von Rohrbach," *J. Reine Angew. Math.*, v. 286/287, 1976, pp. 239–247.

2. G. Hofmeister, "Asymptotische Abschätzungen für dreielementige Extremalbasen in natürlichen Zahlen," *J. Reine Angew. Math.*, v. 232, 1968, pp. 77–101.

3. G. Hofmeister, "Zum Reichweitenproblem bei fester Elementeanzahl." (To appear.)

4. W. F. Lunnon, "A postage stamp problem," *Comput. J.*, v. 12, 1969, pp. 377–380.

5. A. Mrose, "Untere Schränken für Extremalbasen fester Ordnung, I," *Abh. Math. Sem. Univ. Hamburg*, v. 48, 1979, pp. 118–124.

6. B. P. Phillips, Correspondence, *Comput. J.*, v. 19, 1976, p. 93.

7. J. Riddell & C. Chan, "Some extremal 2-bases," *Math. Comp.*, v. 32, 1978, pp. 630–634.

8. H. Rohrbach, "Ein Beitrag zur additiven Zahlentheorie," *Math. Z.*, v. 42, 1937, pp. 1–30.

9. J. L. Seldon, Correspondence, *Comput. J.*, v. 15, 1972, p. 361.

10. E. S. Selmer, "On the postage stamp problem with three stamp denominations," *Math. Scand.*, v. 47, 1980, pp. 29–71.

11. A. Stöhr, "Gelöste und ungelöste Fragen über Basen der natürlichen Zahlenreihe I, II," *J. Reine Angew. Math.*, v. 194, 1955, pp. 40–65, 111–140.