On the expressiveness of subset-sum representations

Lucian Ilie
Arto Salomaa

Turku Centre for Computer Science
TUCS Technical Report No 94
February 1997
ISBN 951-650-964-9
ISSN 1239-1891

Abstract

We develop a general theory for representing information as sums of elements in a subset of the basic set A of cardinality n, often refered to as a "knapsack vector". How many numbers can be represented in this way depends heavily on A. The lower (resp. upper) bound for the cardinality of the set of representable numbers is quadratic (resp. exponential) in terms of n. Our main result is a quadratic-time algorithm for the construction of a knapsack vector of any prescribed expressiveness (that is, the cardinality of the set of representable numbers), provided it falls within the range possible for expressiveness.

1 Introduction

Consider the following very general situation. As a basis we have a finite set A. We will assume that A consists of positive integers but we actually need only the assumption that a commutative and an associative operation $+$ is defined on A. Then each subset S_A of A represents a number, namely, the sum of the elements of S_A. If A has n elements, then at most 2^n numbers can be represented in this way. However, for specific A's this upper bound is often not reached as different subsets S_A may give rise to the same number.

Such subset-sum representations have been often discussed in the literature (see [BF], [To], [Ha], [An]). We mention here only their crucial role in the initial stages of public-key cryptography, [Ad], [MH], [Sa1], [Sa2], [Sa3], [Sh1], [Sh2], [Sh3]. It is customary to refer to the basic set A as a *knapsack vector*. By its *expressiveness* we mean the cardinality of the set it represents. Thus, expressiveness can be at most 2^n. Examples are well known, [MH], where the maximal expressiveness is reached.

However, very little is known about expressiveness in general – a fact we found rather surprising. Number-theoretic studies about partitions cannot be used here. The purpose of this paper is to present the basic facts about expressiveness.

After giving the definitions in Section 2, we discuss the maximal and minimal expressiveness in Sections 3 – 5. We also obtain a very simple characterization of knapsack vectors with minimal expressiveness. Our main result is a quadratic-time algorithm, given in Section 6, for constructing a knapsack vector of any pregiven size and expressiveness, provided the latter falls into the possible range. The range extends from quadratic to exponential. We hope to return to a further study concerning this borderline between polynomial and exponential cases, as well as to the interconnection with the density of knapsack vectors.

2 Definitions

A *knapsack vector* is a vector $A = (a_1, a_2, \ldots, a_n)$ with $n \geq 1$ integer components such that for any $i, 1 \leq i \leq n$, $a_i \geq 1$ and for any $i, j, 1 \leq i, j \leq n$, $i \neq j$ implies $a_i \neq a_j$. An *instance of a knapsack problem* is a pair (A, x) where $A = (a_1, a_2, \ldots, a_n)$ is a knapsack vector and x is a nonnegative integer. A *solution* for (A, x) is a set $S \subseteq \{1, 2, \ldots, n\}$ such that

$$\sum_{i \in S} a_i = x.$$

For a given knapsack vector $A = (a_1, a_2, \ldots, a_n), n \geq 1$, we denote by $Range(A)$ the set of all nonnegative integers x for which the problem (A, x) has at least one solution, that is,

$$Range(A) = \{\sum_{i \in S} a_i \mid S \subseteq \{1, 2, \ldots, n\}\}.$$

It is understood that $\sum_{i \in \emptyset} a_i = 0$, so always $0 \in Range(A)$.

The number of all nonnegative integers x such that (A, x) has a solution is called the *expressiveness* of A and it is denoted

$$Expr(A) = card(Range(A)).$$

For any $n \geq 1$, the set of all knapsack vectors with n components is denoted by $\mathcal{K}_n$. We use also the following notations: the set of all possible values for the expressiveness of knapsack vectors in $\mathcal{K}_n$ is

$$Expr_n = \{Expr(A) \mid A \in \mathcal{K}_n\}$$

and the *minimal* (resp. *maximal*) *expressiveness* of knapsack vectors in $\mathcal{K}_n$ is

$$\begin{aligned} m_n &= \min_{A \in \mathcal{K}_n} Expr(A) = \min(Expr_n), \\ M_n &= \max_{A \in \mathcal{K}_n} Expr(A) = \max(Expr_n), \end{aligned}$$

respectively.

Notice that, since for all $n \geq 1$ the number of subsets of the set $\{1, 2, \ldots, n\}$ is finite, all notions introduced so far are well defined.

3 Maximal expressiveness and injective vectors

In this section, we characterize all vectors possessing the maximal expressiveness M_n. The result in Lemma 3.1 is essentially due to [MH]. It is present here for the sake of completeness and to aid the readability of the rest of the paper. Our argument for Lemma 3.1 appears in a modified form later in the paper.

Obviously, for a knapsack vector with n components $A = (a_1, a_2, \ldots, a_n)$, its expressiveness is at most

$$Expr(A) \leq 2^n$$

that is, the number of all subsets of the set $\{1, 2, \ldots, n\}$.

We show that this bound can be reached.

A knapsack vector $A = (a_1, a_2, \ldots, a_n)$ is called *super-increasing* if for any $i, 2 \leq i \leq n$,

$$a_i > \sum_{j=1}^{i-1} a_j.$$

Lemma 3.1 *If $A = (a_1, a_2, \ldots, a_n)$ is a super-increasing knapsack vector, then*

$$Expr(A) = 2^n.$$

Proof. It is enough to prove that if $x \in Range(A)$ with $x = \sum_{i \in S} a_i$ for some $S \subseteq \{1, 2, \ldots, n\}$, then the subset S is uniquely determined by x. We present the following *algorithm* which gives, for any integer $x \in Range(A)$, the respective subset S.

Algorithm 3.2
1. $i \longleftarrow n$, $S \longleftarrow \emptyset$, $y \longleftarrow x$,
2. if $y \geq a_i$, then $S \longleftarrow S \cup \{i\}$, $y \longleftarrow y - a_i$,
3. $i \longleftarrow i - 1$,
4. if $y > 0$, then go to step 2,
5. output S.

It is straightforward to check that $x = \sum_{i \in S} a_i$. Notice that, if in step 2, $y \geq a_i$, then i must belong to S because otherwise the sum of all a_j's, for $1 \leq j \leq i - 1$, cannot sum up to y. This shows that indeed S is uniquely determined by x. Consequently, any element of the set $Range(A)$ can be obtained in a unique way, hence the number of all element of $Range(A)$ is exactly the number of all subsets of $\{1, 2, \ldots, n\}$, that is, 2^n and the lemma follows.

$\square$

Consequently, we get

Theorem 3.3 *For any $n \geq 1$, $M_n = 2^n$.*

A knapsack vector $A = (a_1, a_2, \ldots, a_n)$ is called *injective* if for any non-negative integer x there is at most one set $S \subseteq \{1, 2, \ldots, n\}$ such that

$$\sum_{i \in S} a_i = x.$$

It is easy to see that

Theorem 3.4 *For any $n \geq 1$, a knapsack vector in $\mathcal{K}_n$ is of maximal expressiveness $M_n = 2^n$ if and only if it is injective.*

4 Minimal expressiveness

We show that, for any $n \geq 1$, the minimal expressiveness of a knapsack vector with n components is given by the vector $V_n = (1, 2, \ldots, n)$. We first compute the number $Expr(V_n)$ and, after that, we prove that indeed it is minimal.

Lemma 4.1 *For any* $n \geq 1$,

$$Expr(V_n) = \frac{n(n+1)}{2} + 1.$$

Proof. It is enough to prove that any integer $x, 0 \leq x \leq \frac{n(n+1)}{2}$ belongs to the set $Range(V_n)$ since $\frac{n(n+1)}{2}$ is the sum of all components of V_n.

For take an arbitrary such x. A set $S \subseteq \{1, 2, \ldots, n\}$ such that $x = \sum_{i \in S} i$ is computed by the *algorithm* below:

Algorithm 4.2
1. $i \longleftarrow n$, $S \longleftarrow \emptyset$, $y \longleftarrow x$,
2. if $y \geq i$, then $S \longleftarrow S \cup \{i\}$, $y \longleftarrow y - i$,
3. $i \longleftarrow i - 1$,
4. if $y > 0$, then go to step 2,
5. output S.

Observe that this algorithm is similar to Algorithm 3.2 in the proof of Lemma 3.1 but they are working under different circumstances: the one in the proof of Lemma 3.1 finds a set S for a nonnegative integer x for which it is known that such a set exists while this one works with an arbitrary x in the mentioned interval.

We prove now by induction on x that the algorithm works as required. If $x = 0$, then $S = \emptyset$ and the answer is correct. Suppose now that for some $x, 0 \leq x \leq \frac{n(n+1)}{2} - 1$, our algorithm works correctly and consider $x + 1$ as input. Consider the largest index, say i_0, such that the choice at step 2 for $x + 1$ differs from the one for x (there must be such an index, otherwise $x = x + 1$). It follows that $y_{(x+1)} \geq i_0$ but $y_{(x)} \not\geq i_0$, where $y_{(z)}$ denotes the variable y for the algorithm on input z. But $y_{(x+1)} = y_{(x)} + 1$, hence $y_{(x)} = i_0 - 1, y_{(x+1)} = i_0$. Now, if $i_0 \geq 2$, then the algorithm for x ends by adding the index $i_0 - 1$ to $S_{(x)}$ while the algorithm for $x + 1$ ends by adding i_0 to $S_{(x+1)}$ ($S_{(z)}$ denotes the variable S for the algorithm on input z) and if $i_0 = 1$, then $y_{(x)} = 0$ and the algorithm for x ends with the current $S_{(x)}$ while the algorithm for $x + 1$ ends by adding 1 to $S_{(x+1)}$.

In the former case we get $S_{(x+1)} = (S_{(x)} - \{i_0 - 1\}) \cup \{i_0\}$ and, using the inductive hypothesis,

$$x + 1 = \sum_{i \in S_{(x)}} i + 1 = \sum_{i \in S_{(x+1)}} i + (i_0 - 1) - i_0 + 1 = \sum_{i \in S_{(x+1)}} i.$$

In the latter case we have $S_{(x+1)} = S_{(x)} \cup \{1\}$ and, using again the inductive hypothesis, we get

$$x + 1 = \sum_{i \in S_{(x)}} i + 1 = \sum_{i \in S_{(x+1)}} i - 1 + 1 = \sum_{i \in S_{(x+1)}} i.$$

Consequently, in both cases, the algorithm works correctly and our result is proved.

$\square$

Lemma 4.3 *For any knapsack vector $A = (a_1, a_2, \ldots, a_n)$,*

$$Expr(A) \geq \frac{n(n+1)}{2} + 1.$$

Proof. Consider a knapsack vector $A = (a_1, a_2, \ldots, a_n), n \geq 1$. Obviously, for any permutation $\pi \in \mathcal{S}_n$, the vector $A_\pi = (a_{\pi(1)}, a_{\pi(2)}, \ldots, a_{\pi(n)})$ is a knapsack vector and

$$Expr(A) = Expr(A_\pi).$$

Thus, we may suppose that A is in increasing order, i.e. $a_1 < a_2 < \cdots < a_n$.

Let us consider the following strictly increasing sequence of integers:

$$0 < \underbrace{a_1 < a_2 < \cdots < a_n}_{n} <$$

$$< \underbrace{a_n + a_1 < a_n + a_2 < \cdots < a_n + a_{n-1}}_{n-1} <$$

$$< \underbrace{a_n + a_{n-1} + a_1 < a_n + a_{n-1} + a_2 < \cdots < a_n + a_{n-1} + a_{n-2}}_{n-2} <$$

$$< \underbrace{a_n + a_{n-1} + a_{n-2} + a_1 < \cdots < a_n + a_{n-1} + a_{n-2} + a_{n-3}}_{n-3} <$$

$$< \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots <$$

$$< \underbrace{a_n + a_{n-1} + \cdots + a_3 + a_1 < a_n + a_{n-1} + \cdots + a_3 + a_2}_{2} <$$

$$< \underbrace{a_n + a_{n-1} + \cdots + a_2 + a_1}_{1}.$$

Obviously, any of the integers in the sequence above belongs to the set $Range(A)$. Since there are

$$1 + n + n - 1 + n - 2 + \cdots + 2 + 1 = \frac{n(n+1)}{2} + 1,$$

(distinct) numbers, it follows that

$$Expr(A) \geq \frac{n(n+1)}{2} + 1$$

as claimed.

$\square$

From Lemmas 4.1 and 4.3 we get the minimal expressiveness for knapsack vectors with n components:

Theorem 4.4 *For any $n \geq 1$, $m_n = \frac{n(n+1)}{2} + 1$.*

5 Knapsack vectors of minimal expressiveness

We want to characterize all knapsack vectors which realize the minimal expressiveness. We use the proof of Lemma 4.3.

Lemma 5.1 *If $A = (a_1, a_2, \ldots, a_n) \in \mathcal{K}_n, n \geq 2$, such that*

$$Expr(A) = \frac{n(n+1)}{2} + 1$$

then

$$a_i = ia_1, \text{ for any } i, 2 \leq i \leq n.$$

Proof. Take $A = (a_1, a_2, \ldots, a_n)$ a knapsack vector of minimal expressiveness. We may again assume that the elements of A are in increasing order. Consider the strictly increasing sequence of $\frac{n(n+1)}{2} + 1$ nonnegative integers in Lemma 4.3. Since

$$Expr(A) = \frac{n(n+1)}{2} + 1,$$

it follows that those numbers are exactly the set $Range(A)$, that is,

$$Range(A) = \{0\} \cup \{a_i \mid 1 \leq i \leq n\} \cup \{\sum_{i \in \{k, k+1, \ldots, n\} \cup \{l\}} a_i \mid 2 \leq k \leq n, 1 \leq l \leq k-1\}.$$

Consider now the sum $a_{n-1} + a_1$. Because $a_{n-1} + a_1 \in Range(A)$ and $a_{n-1} < a_{n-1} + a_1 < a_n + a_1$, it follows that we must have $a_{n-1} + a_1 = a_n$. Take now $a_{n-1} + a_2$. Again, $a_{n-1} + a_2 \in Range(A)$ and

$$a_n = a_{n-1} + a_1 < a_{n-1} + a_2 < a_n + a_2$$

implies $a_{n-1} + a_2 = a_n + a_1$ and so $a_2 = 2a_1$.

Suppose now inductively that, for some $i, 3 \leq i \leq n$, we have $a_j = ja_1$, for all $j, 2 \leq j \leq i-1$, and let us prove that $a_i = ia_1$. Consider the sum $a_{n-1} + a_i \in Range(A)$. As above,

$$a_n + a_{i-2} = a_{n-1} + (i-1)a_1 = a_{n-1} + a_{i-1} < a_{n-1} + a_i < a_n + a_i,$$

thus

$$a_{n-1} + a_i = a_n + a_{i-1} = a_{n-1} + a_1 + (i-1)a_1.$$

Consequently, $a_i = ia_1$ and our result is proved.

□

From Lemma 5.1 we get the characterization of all vectors of minimal expressiveness.

Theorem 5.2 *The set of all knapsack vectors with $n \geq 1$ components for which the minimal expressiveness is reached is*

$$\{(k, 2k, 3k, \ldots, nk) \mid k \geq 1\}.$$

6 On the values of expressiveness

As seen in the previous sections, for a given $n \geq 1$, the expressiveness of all knapsack vectors with n components ranges between $m_n = \frac{n(n+1)}{2} + 1$ (Theorem 4.4) and $M_n = 2^n$ (Theorem 3.3). We show in this section that, rather unexpectedly, for any $n \geq 1$, the expressiveness of knapsack vectors of dimension n can take all values between the two optimal bounds above. That is, we intend to prove now our main result.

Theorem 6.1 *For any $n \geq 1$,*

$$Expr_n = \{\frac{n(n+1)}{2} + 1, \frac{n(n+1)}{2} + 2, \ldots, 2^n\}.$$

The proof of Theorem 6.1 will be essentially contained in the following two lemmas.

Lemma 6.2 *For any integers $n \geq 3$ and $\frac{n(n+1)}{2} + 1 \leq m \leq 2(n-2)(n-1)+4$, there exists a knapsack vector with n components and of expressiveness m.*

Proof. Take $n \geq 3$ and consider the knapsack vectors

$$A_{n,k,l} = (1, 2, 3, \ldots, n-2, k, l),$$

for any

$$\begin{aligned} n-1 \quad &\le k \le \quad \frac{(n-2)(n-1)}{2}+1, \\ k+1 \quad &\le l \le \quad \frac{(n-2)(n-1)}{2}+k+1. \end{aligned} \tag{1}$$

We claim that

$$Expr(A_{n,k,l}) = \frac{(n-2)(n-1)}{2}+k+l+1. \tag{2}$$

It obviously suffices to show that

$$Range(A_{n,k,l}) = \{0,1,2,\dots,\frac{(n-2)(n-1)}{2}+k+l\}. \tag{3}$$

We prove the last equality by showing inclusion in both directions. The part " $\subseteq$ " is obvious since the sum of all components of $A_{n,k,l}$ is $\frac{(n-2)(n-1)}{2}+k+l$. For the converse, " $\supseteq$ ", take an integer $p, 0 \le p \le \frac{(n-2)(n-1)}{2}+k+l$. If $p \le \frac{(n-2)(n-1)}{2}$, then, using Algorithm 4.2 in Lemma 4.1, we get a set $S \subseteq \{1,2,\dots,n-2\}$ such that $\sum_{i\in S} i = p$. Now, $S \subseteq \{1,2,\dots,n\}$ and

$$\sum_{i\in S}(A_{n,k,l})_i = p,$$

where $(A_{n,k,l})_i$ denotes the ith component of $A_{n,k,l}$. If $p \ge \frac{(n-2)(n-1)}{2}+1$, then

$$0 \le p-k \le \frac{(n-2)(n-1)}{2}+l$$

and we have either $0 \le p-k \le \frac{(n-2)(n-1)}{2}$ or else $p \ge \frac{(n-2)(n-1)}{2}+k+1$ in which case

$$0 \le p-k-l \le \frac{(n-2)(n-1)}{2}.$$

Consequently, one of the numbers $p_1 = p-k, p_2 = p-k-l$, say $p_j, j \in \{1,2\}$, is between 0 and $\frac{(n-2)(n-1)}{2}$ and, using again Algorithm 4.2, we get $S' \subseteq \{1,2,\dots,n-2\}$ such that $\sum_{i\in S'} i = p_j$. But now, for $S = S' \cup S_j$, where $S_1 = \{n-1\}, S_2 = \{n-1,n\}$, we have

$$\sum_{i\in S}(A_{n,k,l})_i = \sum_{i\in S'} i + \sum_{i\in S_j}(A_{n,k,l})_i = p$$

so equality (3) is proved and (2) follows.

Take now an integer $m, \frac{n(n+1)}{2}+1 \le m \le 2(n-2)(n-1)+4$. Suppose first that $m \le (n-2)(n-1)+2n$ and choose

$$\begin{aligned} k &= n-1, \\ l &= m-n-\frac{(n-2)(n-1)}{2}. \end{aligned}$$

It follows that

$$k+1=n\leq l\leq \frac{(n-2)(n-1)}{2}+n=\frac{(n-2)(n-1)}{2}+k+1$$

so l is in the allowed interval (1). (This is obvious for k.) We get, by (2),

$$Expr(A_{n,k,l})=m.$$

Consider now the case $m\geq (n-2)(n-1)+2n+1$ and take

$$k=\frac{m-(n-2)(n-1)-even(m)-1}{2},$$

$$l=\frac{m+even(m)-1}{2},$$

where

$$even(m)=\begin{cases}1, & \text{if } m \text{ is even,}\\ 0, & \text{otherwise.}\end{cases}$$

It can then be computed

$$n-1\leq n-\frac{even((n-2)(n-1)+2n+1)}{2}\leq k\leq$$
$$\leq \frac{(n-2)(n-1)}{2}+\frac{1-even(2(n-2)(n-1)+4)}{2}+1=\frac{(n-2)(n-1)}{2}+1$$

so the choice for k is correct with respect to (1). For l we have

$$l=k+\frac{(n-2)(n-1)}{2}+even(m)$$

which obviously respects (1). Using again (2), we have for the current choices

$$Expr(A_{n,k,l})=m$$

and the lemma is proved.

$\square$

Lemma 6.3 *For any integers $n\geq 5$, $2\leq i\leq n-3$, and $2^{n-i-2}i(i+1)+2^{n-i-1}+3\leq m\leq 2^{n-i-1}i(i+1)+2^{n-i}$, there exists a knapsack vector with n components and of expressiveness m.*

Proof. For any fixed $n\geq 5, 2\leq i\leq n-3$, consider the knapsack vectors

$$A_{n,i,k,l}=(1,2,\ldots,i,\tfrac{1}{2}i(i+1)+1,i(i+1)+2,2i(i+1)+2^2,$$
$$2^2i(i+1)+2^3,\ldots,2^{n-i-4}i(i+1)+2^{n-i-3},k,l)$$

for any

$$\begin{aligned} 2^{n-i-4}i(i+1)+2^{n-i-3}+1 &\leq k \leq 2^{n-i-3}i(i+1)+2^{n-i-2}, \\ k+1 &\leq l \leq 2^{n-i-3}i(i+1)+2^{n-i-2}+k. \end{aligned} \tag{4}$$

We claim that

$$Expr(A_{n,i,k,l}) = 2^{n-i-3}i(i+1)+2^{n-i-2}+k+l. \tag{5}$$

We prove this by showing that

$$Range(A_{n,i,k,l}) = \{0,1,2,\ldots,2^{n-i-3}i(i+1)+2^{n-i-2}+k+l-1\}.$$

We point out only the fact that any component $(A_{n,i,k,l})_j$, $i+1 \leq j \leq n-2$, is one unit larger than the sum of all previous components and the same holds true for the maximal values, according to (4), of the last two components. The equality now follows by an iteration of an argument similar to the one in the proof of Lemma 6.2.

Take now m with $2^{n-i-2}i(i+1)+2^{n-i-1}+3 \leq m \leq 2^{n-i-1}i(i+1)+2^{n-i}$. We consider again two cases. Firstly, if $m \leq 3\cdot 2^{n-i-3}i(i+1)+3\cdot 2^{n-i-2}+2$, we chose

$$\begin{aligned} &k = 2^{n-i-4}i(i+1)+2^{n-i-3}+1, \\ &l = m-3\cdot 2^{n-i-4}i(i+1)-3\cdot 2^{n-i-3}-1. \end{aligned}$$

We have that

$$\begin{gathered} k+1 = 2^{n-i-4}i(i+1)+2^{n-i-3}+2 \leq l \leq \\ \leq 3\cdot 2^{n-i-4}i(i+1)+3\cdot 2^{n-i-3}+1 = 2^{n-i-3}i(i+1)+2^{n-i-2}+k \end{gathered}$$

so the choice of l is according to (4). By (5), we have

$$Expr(A_{n,i,k,l}) = m.$$

For the other case, that is, $m \geq 3\cdot 2^{n-i-3}i(i+1)+3\cdot 2^{n-i-2}+3$, we take

$$\begin{aligned} &k = \frac{m-2^{n-i-2}i(i+1)-2^{n-i-1}+odd(m)}{2}, \\ &l = \frac{m-odd(m)}{2} \end{aligned}$$

for

$$odd(m) = \begin{cases} 1, & \text{if } m \text{ is odd,} \\ 0, & \text{otherwise.} \end{cases}$$

Now, we can compute

$$
\begin{gathered}
2^{n-i-4}i(i+1)+2^{n-i-3}+1 \leq \\
\leq 2^{n-i-4}i(i+1)+2^{n-i-3}+1+\frac{1+odd(3\cdot 2^{n-i-3}i(i+1)+3\cdot 2^{n-i-2}+3)}{2} \leq \\
\leq k \leq 2^{n-i-3}i(i+1)+2^{n-i-2}+\frac{odd(2^{n-i-1}i(i+1)+2^{n-i})}{2} = \\
= 2^{n-i-3}i(i+1)+2^{n-i-2}
\end{gathered}
$$

and so the choice of k respects (4). For l we have

$$l = 2^{n-i-3}i(i+1)+2^{n-i-2}+k-odd(m)$$

and it can be easily seen that also the choice of l falls within the proper range.

We obtain, again by (5), that for our choices

$$Expr(A_{n,i,k,l}) = m$$

and the proof is concluded.

$\square$

Proof of Theorem 6.1. For $n \in \{1,2,3\}$, there is nothing to prove since $\frac{n(n+1)}{2}+1 \geq 2^n-1$ for any of these values. Suppose then $n \geq 4$. By Lemma 6.2 we have that

$$B_{n-2} = \{\frac{n(n+1)}{2}+1, \frac{n(n+1)}{2}+2, \ldots, 2(n-2)(n-1)+4\} \subseteq Expr_n. \quad (6)$$

Because, for $n=4$, we have $2(n-2)(n-1)+4 = 2^n$, this case is also proved. So, we may suppose $n \geq 5$. Now, by Lemma 6.3, for any $i, 2 \leq i \leq n-3$,

$$B_i = \{g_n(i), g_n(i)+1, \ldots, f_n(i)\} \subseteq Expr_n \quad (7)$$

where we have denoted $g_n(i) = 2^{n-i-2}i(i+1)+2^{n-i-1}+3$ and $f_n(i) = 2^{n-i-1}i(i+1)+2^{n-i}$. It can be checked that

$$g_n(n-3) \leq 2(n-2)(n-1)+4,$$

for any $n \geq 3$, and that

$$g_n(i-1) \leq f_n(i)$$

for any $n \geq 6, 3 \leq i \leq n-3$. It follows that

$$\{\frac{n(n+1)}{2}+1, \frac{n(n+1)}{2}+2, \ldots, 2^n\} \subseteq \bigcup_{i=2}^{n-2} B_i \quad (8)$$

since $f_n(2) = 2^n$. Finally, using (6),(7), and (8), we get

$$Expr_n = \{\frac{n(n+1)}{2} + 1, \frac{n(n+1)}{2} + 2, \ldots, 2^n\}$$

as claimed.

$\square$

From the proofs of Lemmas 6.2 and 6.3 we get the following quadratic-time algorithm which computes a knapsack vector with a given number of components (at least three, otherwise is trivial) and of a given expressiveness (between the two bounds above).

Algorithm 6.4

Input: *integers n, m such that $n \geq 3$, $\frac{n(n+1)}{2} + 1 \leq m \leq 2^n$.*

Output: *$A \in \mathcal{K}_n$ with $Expr(A) = m$.*

1. *If $m \geq (n-2)(n-1) + 2n + 1$ then go to step 2, else output*

$$A = (1, 2, 3, \ldots, n-2, n-1, m - n - \frac{(n-2)(n-1)}{2}).$$

2. *If $m \geq 2(n-2)(n-1) + 5$ then go to step 3, else output*

$$A = (1, 2, 3, \ldots, n-2, \frac{m - (n-2)(n-1) - even(m) - 1}{2}, \frac{m + even(m) - 1}{2}).$$

3. $i \longleftarrow n - 3$

4. *If $m \leq 2^{n-i-1} i(i+1) + 2^{n-i}$ then go to step 5, else $i \longleftarrow i - 1$; go to step 4.*

5. *If $m \geq 3 \cdot 2^{n-i-3} i(i+1) + 3 \cdot 2^{n-i-2} + 3$ then go to step 6, else output*

$$A = (1, 2, \ldots, i, \tfrac{1}{2} i(i+1) + 1, i(i+1) + 2, 2i(i+1) + 2^2, \ldots, 2^{n-i-4} i(i+1) + 2^{n-i-3}, 2^{n-i-4} i(i+1) + 2^{n-i-3} + 1, m - 3 \cdot 2^{n-i-4} i(i+1) - 3 \cdot 2^{n-i-3} - 1).$$

6. *Output*

$$A = (1, 2, \ldots, i, \tfrac{1}{2} i(i+1) + 1, i(i+1) + 2, 2i(i+1) + 2^2, 2^2 i(i+1) + 2^3, \ldots, 2^{n-i-4} i(i+1) + 2^{n-i-3}, \frac{m - 2^{n-i-2} i(i+1) - 2^{n-i-1} + odd(m)}{2}, \frac{m - odd(m)}{2}).$$

7 Ambiguity and injectivity

Another important topic that should be clarified concerns *ambiguity*. We just indicate here the basic definitions.

For a knapsack vector $A = (a_1, a_2, \dots, a_n)$ and a nonnegative integer $x \geq 0$, the *ambiguity of x with respect to A* is, by definition,

$$Amb_A(x) = card\{S \mid S \subseteq \{1, 2, \dots, n\}, \sum_{i \in S} a_i = x\},$$

that is, the number of possibilities of finding x as a sum of components of A. (Obviously, $x \in Range(A)$ iff $Amb_A(x) \geq 1$.)

For a given knapsack vector $A \in \mathcal{K}_n$, the *ambiguity of A* is the maximal ambiguity w.r.t. A of integers in $Range(A)$, that is,

$$Amb(A) = \max_{x \in Range(A)} Amb_A(x).$$

The vector A is *unambiguous* if and only if $Amb(A) = 1$. Clearly, A is unambiguous if and only if A is injective.

Finally, we define the *maximal ambiguity* of knapsack vectors with $n \geq 1$ components by

$$Amb_n = \max_{A \in \mathcal{K}_n} Amb(A).$$

Also very little is known about algorithms dealing with injectivity. Consider the decision problems formulated as follows:

KNAPSACK

Input: a knapsack vector $A = (a_1, a_2, \dots, a_n)$ and a positive integer x.
Question: is there a subset $S \subseteq \{1, 2, \dots, n\}$ such that $\sum_{i \in S} a_i = x$?

KNAPSACK$_{\text{INJ}}$

Input: an injective knapsack vector $A = (a_1, a_2, \dots, a_n)$ and a positive integer x.
Question: is there a subset $S \subseteq \{1, 2, \dots, n\}$ such that $\sum_{i \in S} a_i = x$?

INJECTIVITY

Input: a knapsack vector $A = (a_1, a_2, \dots, a_n)$.
Question: is A injective ?

Theorem 7.1 KNAPSACK$_{\text{INJ}}$ *is polynomial-time reducible to* co-INJECTIVITY.

Proof. Consider an instance of $\mathrm{KNAPSACK_{INJ}}$: an injective knapsack vector $A = (a_1, a_2, \ldots, a_n)$ and a positive integer x. We construct the following instance of co-INJECTIVITY:

$$B = (b_1, b_2, \ldots, b_{2n+2}),$$

where

$$\begin{aligned}
&\left.\begin{aligned} b_{2i-1} &= a_i 2^{n+1} + 2^i, \\ b_{2i} &= a_i 2^{n+1} - 2^i, \end{aligned}\right\} \text{ for any } 1 \le i \le n, \\
&b_{2n+1} = (\sum_{i=1}^{n} a_i)2^{n+1} - x2^{n+1} + \sum_{i=1}^{n} 2^i + (\sum_{i=1}^{n} a_i)2^{n+2}, \\
&b_{2n+2} = x2^{n+1} + (\sum_{i=1}^{n} a_i)2^{n+2}.
\end{aligned}$$

Notice first that the above transformation can be easily done by a Turing machine operating in polynomial-time (even in logarithmic space).

Claim. The knapsack problem (A, x) has a solution if and only if the knapsack vector B is not injective.

Proof of Claim. Suppose first that (A, x) has a solution. So, for some $S \subseteq \{1, 2, \ldots, n\}, \sum_{i \in S} a_i = x$. It can be checked now that for

$$\begin{aligned}
S_1 &= \{2i \mid i \in S\} \cup \{2n+1\}, \\
S_2 &= \{2i-1 \mid i \in \{1, 2, \ldots, n\} \backslash S\} \cup \{2n+2\},
\end{aligned}$$

we have

$$\sum_{i \in S_1} b_i = \sum_{i \in S_2} b_i. \tag{9}$$

and so B is not injective.

Conversely, suppose that B is not injective. Thus, there are two disjoint subsets, say S_1, S_2, of $\{1, 2, \ldots, n\}$ such that (9) holds.

We prove first that at least one of the indices $2n+1$ and $2n+2$ appears in the set $S_1 \cup S_2$.

If this is not the case, then $S_1 \cup S_2 \subseteq \{1, 2, \ldots, n\}$ and, for any $1 \le i \le n$, $j \in \{1, 2\}$, if the index $2i-1$ or $2i$ belongs to S_j, then the other one must be in S_j too, because of the term 2^i (or -2^i, resp.). To see this, notice first that, clearly, it is not possible that all indices in $S_1 \cup S_2$ are odd and consider the smallest i such that $2i \in S_1 \cup S_2$; call it i_0. Suppose that $2i_0 - 1 \notin S_1 \cup S_2$. The binary representation of b_{2i_0} ends with

$$\ldots \underbrace{11\ldots1}_{n+1-i_0}\underbrace{00\ldots0}_{i_0}.$$

But now, all the numbers $\{b_i \mid j \in S_1 \cup S_2 \backslash \{2i_0\}\}$ have in all the last $i_0 + 1$ positions in the binary representation only 0's and the digit in the $(i_0 + 1)$st position (from the right) in the binary representation of b_{2i_0} is 1 thus the equality (9) is impossible, a contradiction.

It follows that for

$$S'_j = \{i \in \{1, 2, \dots, n\} \mid 2i - 1, 2i \in S_j\}, j = 1, 2,$$

we have, by (9),

$$\sum_{i \in S'_1} (a_i 2^{n+1} + 2^i + a_i 2^{n+1} - 2^i) = \sum_{i \in S'_2} (a_i 2^{n+1} + 2^i + a_i 2^{n+1} - 2^i)$$

and therefore

$$\sum_{i \in S'_1} a_i = \sum_{i \in S'_2} a_i. \tag{10}$$

Since $S'_1 \cap S'_2 = \emptyset$ and $S'_1 \cup S'_2 \neq \emptyset$, this contradicts the injectivity of A.

It follows that at least one of $2n + 1$ and $2n + 2$ appears in $S_1 \cup S_2$. Now, due to the large number $(\sum_{i=1}^{n} a_i) 2^{n+2}$ which appears as a term in each of b_{2n+1} and b_{2n+2}, we get that both $2n+1$ and $2n+2$ must apper and, moreover, one in S_1 and the other in S_2.

Suppose that $2n + 1 \in S_1$. As above, the term $\sum_{i=1}^{n} 2^i$ appearing in b_{2n+1} cannot be equalized otherwise than taking, for each $1 \leq i \leq n$, exactly one of $2i - 1$ and $2i$ into $S_1 \cup S_2$ and, moreover, all even indices must be taken into S_1 while all the odd ones into S_2. If we put $S = \{i \in \{1, 2, \dots, n\} \mid 2i \in S_1\}$, then we easily compute from (10) that

$$\sum_{i \in S} a_i = x,$$

thus the problem (A, x) has a solution, as claimed.

□

While it is well known that KNAPSACK is NP-complete, it seems that the completeness of $\text{KNAPSACK}_{\text{INJ}}$ is open. By Theorem 7.1, the completeness of $\text{KNAPSACK}_{\text{INJ}}$ would imply that INJECTIVITY is coNP-complete, since it is obviously in coNP.

References

[Ad] L. Adleman, On breaking the iterated Merkle-Hellman public-key cryptosystem, Proceedings of the 15th ACM Symposium on the Theory of Computing (1983) 402–412.

[An] I. Anderson, Combinatorics of finite sets, (Oxford Univ. Press, New York, 1987).

[BF] G. Bergman, K. D. Fryer, Introduction to combinatorics, (Academic Press, New York, 1972).

[Ha] M. Hall, Combinatorial theory, (Wiley & Sons, New York, 1986).

[MH] R. Merkle, M. Hellman, Hiding information and signatures in trapdoor knapsacks, IEEE Transactions on Information Theory, IT–24 (1978) 525–530.

[Sa1] A. Salomaa, A deterministic algorithm for modular knapsack problems, Theoretical Computer Science 88(1) (1991) 127–138.

[Sa2] A. Salomaa, Decision problems arising from knapsack transformations, Acta Cybernetica 9(4) (1990) 419–440.

[Sa3] A. Salomaa, Public-Key Cryptography, (Springer Verlag, Berlin, 1990).

[Sh1] A. Shamir, A fast signature scheme, MIT Laboratory for Computer Science, Tehnical Report, 1978.

[Sh2] A. Shamir, Embedding cryptographic trapdoor in arbitrary knapsack systems, MIT Laboratory for Computer Science, Tehnical Report 230, 1982.

[Sh3] A. Shamir, A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem, Proceedings of the 23rd FOCS Symposium (1982) 145 – 152.

[To] I. Tomescu, Problems in combinatorics and graph theory, (Wiley & Sons, New York, 1985).

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

http://www.tucs.abo.fi

University of Turku
- **Department of Mathematical Sciences**

Åbo Akademi University
- **Department of Computer Science**
- **Institute for Advanced Management Systems Research**

Turku School of Economics and Business Administration
- **Institute of Information Systems Science**