A deterministic algorithm for modular knapsack problems

Arto Salomaa
Mathematics Department, University of Turku, 20500 Turku, Finland

Communicated by G. Rozenberg
Received May 1989

Abstract

We present an algorithm for finding out whether or not a given vector results from some super-increasing vector by modular multiplication. In the positive case the algorithm produces such a super-increasing vector as well as a multiplier and a modulus. The algorithm is deterministic and is based on elementary considerations only.

1. Connection with cryptography

Being an especially lucid and easy-to-explain example of an NP-complete problem, the knapsack problem appears in many different contexts. For instance, it is the core of the first public-key cryptosystem [1]. One starts with a super-increasing knapsack vector A. Knapsack problems based on a super-increasing vector are easy to solve; the algorithm consists of just one sweep from right to left. The super-increasing vector A is scrambled by modular multiplication, and the resulting vector \( B = (b_1, \ldots, b_n) \) is publicized as the encryption key. A plain-text block of \( n \) bits, viewed as a column vector \( T \), is encrypted as the number \( BT \). The trapdoor known to the legal recipient consists of the inverse multiplier and the modulus. Thus, to decrypt, the legal recipient solves a knapsack problem involving a super-increasing \( A \), whereas the eavesdropper has to work with the arbitrary looking \( B \). Details and examples are given in [2].

Shamir [4] gave an algorithm for breaking the cryptosystem by preprocessing. Given \( B \), the algorithm produces a super-increasing \( A' \) (not necessarily the \( A \) used by the cryptosystem designer) such that \( B \) results from \( A' \) by modular multiplication. The algorithm assumes the existence of \( A \), and works in random polynomial time, where the probability of failure can be made arbitrarily small.
The algorithm based on the theorem in Section 3 assumes no oracle information about the existence of \( A \). Given \( B \), the algorithm tests whether or not \( B \) is reachable from a super-increasing vector by modular multiplication and, in the positive case, produces such a super-increasing vector as well as a multiplier and a modulus. The algorithm is deterministic. All details can be presented in a fairly simple way ab ovo, i.e. no sophisticated results are used as tools. Rudiments of the algorithm were given in [3].

Apart from cryptography, studies dealing with knapsack vectors are of interest also on their own right. In what follows, the interconnection with cryptography will not any more be important. All definitions are given in terms of \( n \)-tuples of positive integers.

### 2. Super-reachability

An ordered \( n \)-tuple of distinct positive integers \( A = (a_1, \ldots, a_n), n \geq 3 \), is referred to as a **knapsack vector** of dimension \( n \). A knapsack vector \( A \) is increasing (super-increasing) iff

\[
a_j > a_{j-1} \left( a_j > \sum_{i=1}^{j-1} a_i \right)
\]

holds for all \( j = 2, \ldots, n \). Clearly, every super-increasing vector is increasing. For a knapsack vector \( A \), we define

\[
\max A = \max \{ a_j | 1 \leq j \leq n \}.
\]

For a positive number \( x \), we denote by \([ x ]\) the integer part of \( x \), i.e. the greatest integer \( \leq x \). For integers \( x \) and \( m \geq 2 \), we denote by \((x, \text{mod } m)\) the least nonnegative remainder of \( x \) modulo \( m \). Clearly,

\[
(x, \text{mod } m) = x - [x/m] \cdot m.
\]

This equation will be often written in the form

\[
(2.1) \quad x = (x, \text{mod } m) + [x/m] \cdot m
\]

A knapsack vector \( B = (b_1, \ldots, b_n) \) is \((A, t, m)-reachable\); in symbols,

\[
(2.2) \quad A \xrightarrow{t, m} B
\]

iff \( A = (a_1, \ldots, a_n) \) is an increasing knapsack vector, \( t \) and \( m \geq 2 \) are positive integers with \((t, m) = 1\) and, furthermore, \( b_i = (ta_i, \text{mod } m) \) for \( i = 1, \ldots, n \). (Here \((t, m)\) denotes the greatest common divisor of \( t \) and \( m \).) The integers \( t \) and \( m \) are referred to as the **multiplier** and the **modulus**, respectively.

Similarly, \( B \) is \((A, t, m)-super-reachable\); in symbols,

\[
(2.3) \quad A \xrightarrow{S, t, m} B
\]
iff \( B \) is \((A, t, m)\)-reachable; moreover, \( A \) is super-increasing and \( m > \sum_{i=1}^{n} a_i \). The vector \( B \) is super-reachable iff (2.3) holds for some triple \((A, t, m)\). Thus, the relation (2.3) implies the relation (2.2) but not vice versa, since two additional conditions have to be fulfilled.

**Remark.** A notion of reachability, analogous to that of super-reachability, does not make much sense because every vector would be reachable. Indeed, an arbitrary \( B = (b_1, \ldots, b_n) \) is \((A, 1, m)\)-reachable, where \( m > \max B \) and

\[
A = (b_1, b_2 + k_2 m, \ldots, b_n + k_m m),
\]

where the integers \( k_i \) are chosen in such a way that the resulting vector will be increasing.

The following lemmas contain facts needed in Section 3.

**Lemma 2.1.** Assume that \( A = (a_1, \ldots, a_n) \) is increasing (super-increasing), \( m \geq 2 \) and \((t, m) - 1\). Then each of the vectors

\[
A_k = (a_1 + k \cdot \lceil ta_1/m \rceil, \ldots, a_n + k \cdot \lceil ta_n/m \rceil), \quad k = 1, 2, \ldots.
\]

is increasing (super-increasing).

**Proof.** Denote

\[
a_{j}^{(k)} = a_j + k \cdot \lceil ta_j/m \rceil, \quad 1 \leq j \leq n, \quad k = 1, 2, \ldots
\]

Assume first that \( A \) is increasing. Let \( i \) and \( k \) be arbitrary, \( 2 < i \leq n \). Then, \( a_{i-1}^{(k)} < a_i^{(k)} \) because

\[
a_{i-1} < a_i \quad \text{and} \quad k \cdot \lceil ta_{i-1}/m \rceil \leq k \cdot \lceil ta_i/m \rceil.
\]

Hence, \( A_k \) is increasing.

Assume next that \( A \) is super-increasing. Choose again arbitrary \( i \) and \( k \). The inequality

\[
\sum_{j=1}^{i-1} a_j^{(k)} < a_i^{(k)}
\]

follows because

\[
\sum_{j=1}^{i-1} a_j < a_i
\]

and

\[
\lceil ta_1/m \rceil + \cdots + \lceil ta_{i-1}/m \rceil \leq \lceil (a_1 + \cdots + a_{i-1})/m \rceil \leq \lceil ta_i/m \rceil.
\]

\( \square \)
Lemma 2.2. Let $A$, $t$, $m$ and $A_k$ be defined as in Lemma 2.1. If a knapsack vector $B$ is $(A, t, m)$-reachable ($(A, t, m)$-super-reachable) then for all $k$, $B$ is also $(A_k, t, m+k t)$-reachable ($(A_k, t, m+k t)$-super-reachable).

Proof. Assume again first that $B=(b_1, \ldots, b_n)$ is $(A, t, m)$-reachable. Consequently,

\[ b_i = (t a_i \mod m) \quad \text{for} \quad i = 1, \ldots, n. \]

Clearly, $(t, m+k t) = 1$ and, by (2.1) and (2.4),

\[ t a_i = b_i + \lfloor t a_i / m \rfloor \cdot m. \]

This implies that

\[ t a_i^{(k)} = b_i + \lfloor t a_i / m \rfloor \cdot m + \lfloor t a_i / m \rfloor \cdot k t = b_i + \lfloor t a_i / m \rfloor (m+k t). \]

Since by (2.4) $b_i < m+k t$, we obtain

\[ (t a_i^{(k)}, \mod (m+k t)) = b_i. \]

By Lemma 2.1, $A_k$ is increasing. Hence, (2.5) implies that $B$ is $(A_k, t, m+k t)$-reachable. Assume next that $B$ is $(A, t, m)$-super-reachable. Hence,

\[ \sum_{i=1}^n a_i < m. \]

According to Lemma 2.1, $A_k$ is super-increasing. The equation (2.5) is established in the same way as above. Hence, to prove that $B$ is $(A, t, m)$-super-reachable, it suffices to show that the new modulus is big enough:

\[ \sum_{i=1}^n a_i^{(k)} < m+k t. \]

But this follows by (2.6) and the definition of $a_i^{(k)}$ because

\[ \sum_{i=1}^n k \cdot \lfloor t a_i / m \rfloor \leq k \lfloor t(a_1 + \cdots + a_n) / m \rfloor \leq k \lfloor t \rfloor = k t. \]

It is an immediate consequence of Lemma 2.2 that every super-reachable vector $B$ can be obtained by modular multiplication from infinitely many super-increasing vectors. Thereby, the modulus changes but the multiplier may be kept unchanged.

Because $(t, m) = 1$, there is an integer $t^{-1}$ such that $tt^{-1} \equiv 1 \mod m$. Such an integer is found fast by Euclid's algorithm and may be chosen from the interval $1 \leq t^{-1} < m$. The number $t^{-1}$ will be referred to as the inverse of $t$.

3. The main result

As in Section 2, we consider triples $(A, t, m)$, where $A$ is increasing, $m \geq 2$ and $(t, m) = 1$. Also the vectors $A_k$ are defined as in Section 2. We now try to replace the
A deterministic algorithm for modular knapsack problems

triple \((A, t, m)\) by the triple \((C, t, p)\), where \(C = A_k\) and \(p = m + kt\) for some \(k\); moreover, \(C = (c_1, \ldots, c_n)\) is super-increasing and \(p > \sum_{i=1}^n c_i\). For the smallest number \(k\) satisfying these conditions (provided such numbers \(k\) exist), the triple \((C, t, p)\) is called the goal of the triple \((A, t, m)\).

If \(A\) itself is super-increasing and \(m > \sum_{i=1}^n a_i\) then the triple \((A, t, m)\) is called its own goal. (If we define \(A_0\) similarly as \(A_k\), \(k \geq 1\), then \(A = A_0\) and 0 is the smallest value of \(k\) satisfying the conditions required.)

The next step is to investigate which of the triples \((A, t, m)\) possess goals. A number \(i, 3 \leq i \leq n\), is termed a violation point of \(A\) iff

\[
\sum_{j=1}^{i-1} a_j < \sum_{j=1}^{i-1} a_j.
\]

Thus, the requirement of \(A\) being super-increasing is violated by the \(i\)th component of \(A\). The number 2 can never be a violation point in this sense because \(A\) is increasing.

Assume now that \(i\) is a violation point of \(A\) and that

\[
[t_1/m] + \cdots + [t_{i-1}/m] < [t_i/m].
\]

Then the smallest integer \(x\) such that

\[
\sum_{j=1}^{i-1} a_j + x \sum_{j=1}^{i-1} [t_j/m] < a_i + x \cdot [t_i/m]
\]

is called the rescuer of \(i\). Explicitly,

\[
x = \left\lfloor \frac{\left( \sum_{j=1}^{i-1} a_j - a_i \right)}{[t_i/m] - \sum_{j=1}^{i-1} [t_j/m]} \right\rfloor + 1.
\]

By (3.1) and (3.2), \(x\) is a positive integer.

Assume that (3.2) holds for every violation point \(i\) of \(A\). Then the rescuer of \(A\) is defined to be the maximum of the rescuers of all violation points \(i\).

Assume finally that

\[
m \leq \sum_{i=1}^n a_i
\]

and that

\[
\sum_{i=1}^n [t_i/m] < t.
\]

Then the smallest number \(y\) such that

\[
\sum_{i=1}^n a_i + y \cdot \sum_{i=1}^n [t_i/m] < m + yt
\]

is called the rescuer of \(m\). Explicitly,

\[
y = \left\lfloor \frac{\left( \sum_{i=1}^n a_i - m \right)}{t - \sum_{i=1}^n [t_i/m]} \right\rfloor + 1.
\]
By (3.4) and (3.5), \( y \) is a positive integer. Observe that if (3.3) ((3.6)) holds for some integer \( x \) (\( y \)) then it holds for all integers \( x > y \) as well. In (3.6) we want to assure that a condition corresponding to (2.7) is satisfied.

For the sake of completeness, we say that 0 is the rescuer of \( A \) if \( A \) has no violation points (i.e., \( A \) is super-increasing) and, furthermore, 0 is the rescuer of \( m \) if (3.4) does not hold.

**Lemma 3.1.** A triple \((A, t, m)\) possesses a goal \((C, t, p)\) iff (3.2) holds for every violation point \( i \) of \( A \); moreover, (3.5) holds if (3.4) holds. Assuming that the latter conditions are satisfied, we have \( C = A_k \) and \( p = m + kt \), where \( k \) is the maximum of the rescuers of \( A \) and \( m \).

**Proof.** Recall first that (3.3) holds also for every integer \( x \) greater than the rescuer of \( i \) and that (3.6) holds also for every integer \( y \) greater than the rescuer of \( m \). This means that \((C, t, p)\) is the goal of \((A, t, m)\). In particular, the choice of \( k \) guarantees that we get the smallest possible \( k \). Thus, the “if”-part of the lemma holds true.

To prove the “only if”-part, assume first that

\[
(3.7) \quad \sum_{j=1}^{i-1} \left[ \frac{ta_j}{m} \right] \geq \left[ \frac{ta_i}{m} \right]
\]

holds for some violation point \( i \) of \( A \). We obtain by (3.1) and (3.7)

\[
\sum_{j=1}^{i-1} a_j + x \cdot \sum_{j=1}^{i-1} \left[ \frac{ta_j}{m} \right] \geq a_i + x \cdot \left[ \frac{ta_i}{m} \right].
\]

This means that \( i \) is a violation point of every vector \( A_x \) and consequently, \((A, t, m)\) possesses no goal.

Assume next that (3.4) holds but

\[
(3.8) \quad \sum_{i=1}^{n} \left[ \frac{ta_i}{m} \right] \geq t.
\]

By (3.4) and (3.8) we infer that

\[
\sum_{i=1}^{n} a_i^{(x)} \geq m + xt
\]

holds for all \( x \). This means that the modulus of every vector \( A_x \) is too small and, consequently, \((A, t, m)\) possesses no goal. \( \Box \)

We are now ready to prove our main result.

**Theorem 3.2.** A knapsack vector \( B=(b_1, \ldots, b_n) \) is super-reachable iff there are \( u \) and \( m, (u, m) = 1, u < m, \max B < m \leq 2 \max B \) and an increasing vector \( A=(a_1, \ldots, a_n) \) such that

\[
(3.9) \quad (ub_i, \mod m) = a_i, \quad 1 \leq i \leq n.
\]
and the triple \((A, t = u^{-1}, m)\) possesses a goal. If it exists, the goal \((C, t, p)\) of \((A, t, m)\) satisfies

\[(3.10) \quad C \rightarrow B.\]

**Proof.** Observe first that (3.9) implies the relation \(A \rightarrow B\) because \(m > \max B\). Hence, the "if"-part of the theorem as well as the relation (3.10) follows by Lemma 2.2 and the definition of the goal.

To prove the "only if"-part, we first establish the following lemma.

**Lemma 3.3.** If \(E = (e_1, \ldots, e_n)\) is super-reachable, there are a super-increasing \(D, m\) and \(t \leq \max E\) such that

\[(3.11) \quad D \rightarrow E.\]

**Proof of Lemma 3.3.** Since \(E\) is super-reachable, (3.11) holds for some \(D, t, m\). We assume without loss of generality that \(t < m\) because, if this is not the case originally, we subtract from \(t\) a suitable multiple of \(m\) without changing anything. (The equation \(t = m\) is not possible because \((t, m) = 1\).

Assume, thus, that (3.11) holds with \(\max E < t < m\). We construct another triple \((D_1, t_1, m_1)\) such that

\[(3.12) \quad D \rightarrow E\]

and \(t_1 < t\). The lemma is then established by repeating (if necessary) this construction.

By definition

\[m_1 = t, \quad t_1 = (-m, \mod t)\]

and

\[D_1 = ([td_1/m], \ldots, [td_n/m]),\]

where \(D = (d_1, \ldots, d_n)\). Clearly, \((t_1, m_1) = 1\). By (3.11) and (2.1) we obtain for \(1 \leq i \leq n\)

\[t_1 \cdot [td_i/m] \equiv e_i - td_i \equiv e_i (\mod t).\]

But because \(t > \max E \geq e_i\),

\[t_1 \cdot [td_i/m], \mod t) = e_i.\]

To prove (3.12), we show first that \(D_1\) is super-increasing. Since \(D\) is super-increasing, we have for \(2 \leq i \leq n\)

\[
\sum_{j=1}^{i-1} td_j/m < td_i/m.
\]
Hence,
\[ (3.13) \sum_{j=1}^{i-1} \lfloor td_j/m \rfloor \leq \lfloor td_i/m \rfloor. \]

Assume that we have equality in (3.13). Then
\[ m \sum_{j=1}^{i-1} \lfloor td_j/m \rfloor = m \lfloor td_i/m \rfloor. \]

Applying (2.1) we obtain
\[ \sum_{j=1}^{i-1} (td_j - e_j) = td_i - e_i \]
and, hence,
\[ e_i - \sum_{j=1}^{i-1} e_j = t \left( d_i - \sum_{j=1}^{i-1} d_j \right). \]

The coefficient of \( t \) is positive because \( D \) is super-increasing. Consequently,
\[ t \leq e_i - \sum_{j=1}^{i-1} e_j < e_i \leq \max E. \]

which contradicts the assumption \( t > \max E \). This shows that we must have a strict inequality in (3.13). We conclude that \( D_1 \) is super-increasing.

Finally, we observe that the new modulus is big enough: since \( m \geq \sum_{i=1}^{n} d_i \), we obtain
\[ t > \sum_{i=1}^{n} t d_i / m \geq \sum_{i=1}^{n} \lfloor t d_i / m \rfloor. \]

**Proof of Theorem 3.2 (continued).** Returning to the proof of the “only if”-part of Theorem 3.2, we assume that \( B \) is super-reachable. By Lemma 3.3, there is a super-increasing \( D = (d_1, \ldots, d_n), t \leq \max B \) and \( p \) such that

\[ (3.14) \quad D \rightarrow B. \]

If \( p \leq 2 \max B \) in (3.14), then the conditions listed in the Theorem are satisfied with \( D = A = C, p = m, u = t^{-1} \). Thus, assume that

\[ (3.15) \quad p > 2 \max B \geq 2t. \]

Consider the vector \( D^{(1)} = (d_1^{(1)}, \ldots, d_n^{(1)}) \) with
\[ d_1^{(1)} = d_i - \lfloor t d_i / p \rfloor, \quad 1 \leq i \leq n. \]

We claim that \( B \) is \( (D^{(1)}, t, p-t) \)-reachable. Observe first that
\[ t d_1^{(1)} = t d_i - t \cdot \lfloor t d_i / p \rfloor = b_i + \lfloor t d_i / p \rfloor + p - \lfloor t d_i / p \rfloor \cdot t \equiv b_i \mod (p-t). \]
By (3.15), \( p-t > \max B \geq b_i \) and, consequently,
\[
(td_i, \mod(p-t)) = b_i.
\]

Clearly, \((t, p-t) = 1\). To prove our claim, we still have to show that \(D^{(1)}\) is increasing. Consider an arbitrary \(i, 1 \leq i \leq n-1\). Since \(D\) is increasing,

\[
(3.16) \quad d_{i+1} = d_i + \alpha \quad \text{for some } \alpha \geq 1.
\]

We assume first that \(\alpha > 1\). Then
\[
(3.17) \quad d_{i+1}^{(1)} = d_i + \alpha - \left[ t(d_i + \alpha)/p \right]
\]
\[
\geq d_i + \alpha - \left( 1 + \left[ td_i/p \right] + \left[ t\alpha/p \right] \right)
\]
\[
= d_i^{(1)} + (\alpha - 1) - \left[ t\alpha/p \right] > d_i^{(1)}.
\]

In (3.17) the first inequality follows because always \([x + y] \leq [x] + [y] + 1\). The second inequality follows because \(t/p < \frac{1}{2}\) by (3.15), and hence,
\[
\left[ t\alpha/p \right] \leq t\alpha/p < \alpha/2.
\]

Assume secondly that \(\alpha = 1\) in (3.15). In this case \([t\alpha/p] = 0\). By (3.17) we infer that either \(d_{i+1}^{(1)} > d_i^{(1)}\), or else
\[
(3.18) \quad \left[ t(d_i + 1)/p \right] = 1 + \left[ td_i/p \right].
\]

Choose an integer \(\beta\) such that \(\beta p \leq t d_i < (\beta + 1)p\). (In fact, \(\beta = \left[ td_i/p \right]\).) By (3.18) \((\beta + 1)p < t(d_i + 1)\).

If \(t d_i < (\beta + \frac{1}{2})p\), we infer, because \(t < p/2\),
\[
 t d_i + t < (\beta + \frac{1}{2}) p + t = (\beta + 1) p + t - p/2 < (\beta + 1) p,
\]
a contradiction. Hence, \(t d_i \geq (\beta + \frac{1}{2}) p\). But now
\[
b_i = t d_i - \beta p \geq p/2.
\]

This implies that \(p \leq 2b_i \leq 2 \max B\), again a contradiction.

We have established our claim. It is important to observe that we used only the fact that \(D\) is increasing, and not the fact that \(D\) is super-increasing. This observation gives the possibility to repeat the above construction whenever necessary. Indeed, if \(p-t > 2 \max B\), we form the vector \(D^{(2)}\) with
\[
d_i^{(2)} = d_i^{(1)} - \left[ t d_i^{(1)}/(p-t) \right]
\]
and conclude exactly as above that \(B\) is \((D^{(2)}, t, p-2t)\)-reachable.

Let \(k\) be the smallest integer such that
\[
p-kt \leq 2 \max B.
\]
(Thus, \(p-(k-1)t > 2 \max B\).) Clearly, \(t \leq \max B < p-kt\). By \(k\) applications of the above construction we conclude that \(B\) is \((D^{(k)}, t, p-kt)\)-reachable. Hence, (3.9) holds for the increasing vector \(D^{(k)} = A, m = p-kt\) and \(u = t^{-1}\). Moreover, we have that
max $B < m \leq 2$ max $B$ and $u < m$. We still have to show that the triple $(D^{(k)}, t, m)$ possesses a goal.

**Lemma 3.4.** The goal of the triple $(D^{(k)}, t, m)$ is the first triple $(D^{(k-1)}, t, m + it)$ in the sequence

$$
(D^{(k)}, t, m), (D^{(k-1)}, t, m + t), \ldots, (D^{(1)}, t, m + (k - 1)t), (D^{(0)}, t, m + kt),
$$

where $D^{(0)} = D$, $m + kt = p$, satisfying the relation

$$
D^{(k-i)}(S, t, m + it) \longrightarrow B, \quad 0 \leq i \leq k.
$$

**Proof.** By (3.14) such a first triple exists in the sequence (3.19). The relation (3.20) may be satisfied already for $i = 0$, i.e. the triple $(D^{(k)}, t, m)$ may be its own goal. This happens if our construction of reducing the modulus $p$ does not introduce any violation points for the new vectors $D^{(j)}$ and also does not make the new modulus too small.

Let the vectors $A_k$ be defined as in Lemma 2.1. Consider the sequence of triples

$$
(A, t, m), (A_1, t, m + t), \ldots, (A_{k-1}, t, m + (k - 1)t), (A_k, t, m + kt).
$$

Lemma 3.4 follows from Lemmas 2.2 and 3.1 if the sequences (3.19) and (3.21) coincide.

As regards the multipliers and moduli, it is obvious that the sequences (3.19) and (3.21) coincide. By definition, $D^{(k)} = A$. We still have to show that

$$
D^{(k-i)} = A_i, \quad 1 \leq i \leq k.
$$

Consider first $D^{(k-1)}$ and $A_1$. By the definitions we infer for $1 \leq i \leq n$

$$
(a^{(1)}_i = a_i + \lfloor iu_i/m \rfloor \quad \text{and} \quad d^{(k-1)}_i = d^{(k)}_i + \lfloor t d^{(k-1)}_i/(m + t) \rfloor).
$$

In (3.23), $a_i = d^{(k)}_i$. We know, furthermore, that

$$
(t a^{(1)}_i, \text{mod}(m + t)) = (t d^{(k-1)}_i, \text{mod}(m + t)).
$$

Hence, $d^{(k-1)}_i$ and $a^{(1)}_i$ can differ only if the absolute value of the difference of the two bracket expressions appearing in (3.23) is a positive multiple of $m + t$. Straightforward size estimations show that this is not the case. We conclude that $D^{(k-1)} = A_1$.

Assuming that $D^{(k-j)} = A_j$, $1 \leq j < k$, we obtain equations corresponding to (3.23):

$$
a^{(j+1)}_i = a^{(j)}_i + \lfloor t a^{(j)}_i/m \rfloor,
$$

$$
d^{(k-j-1)}_i = d^{(k-j)}_i + \lfloor t d^{(k-j-1)}_i/(m + j t + t) \rfloor,
$$

and establish exactly as above that $D^{(k-j-1)} = A_{j+1}$.

We have concluded the proof of Lemma 3.4 and, hence, also the proof of the “only if”-part of Theorem 3.2. □
4. The algorithm and concluding remarks

Given $B$, consider pairs $(u, m)$ such that $(u, m) = 1$, $u < m$, $max B < m \leq 2 \max B$ and $(3.9)$ holds for some increasing $A$. Check that $t = u^{-1} \leq \max B$. (In fact, this is not necessary.) Check that $(3.2)$ is satisfied for each violation point of $A$ and that $(3.5)$ holds if $(3.4)$ holds. In case of a "no"-answer to any of these questions, start with new pair $(u, m)$. If you get only positive answers, the rescuers determine the triple $(D, t, p)$ such that $(3.14)$ holds. $B$ is not super-reachable if all pairs $(u, m)$ give "no"-answers.

In the complexity estimation it is important to notice that much of the above discussions are there only to show the correctness of the algorithm and, thus, do not affect the complexity. The complexity in terms of $n$ depends on the growth of the entries of $B$ with respect to $n$, whereas our algorithm is independent of any growth bounds which always restrict the instances to be considered. If the entries of $B$ are assumed to grow linearly with $n$, there are $O(n^2)$ pairs $(u, m)$. The checking needed for each pair can be performed in $O(n)$ time. This gives the total complexity $O(n^3)$. The linear inequalities dealing with the rescuers have to be solved only once.

The algorithm works often in the case of "composite knapsacks", too. This means that the vector $B$ is obtained from a super-increasing vector by several successive modular multiplications. For instance, applying pairs $(4, 9), (3, 20), (3, 17), (2, 23)$ to the super-increasing vector $(1, 2, 4)$ we get the sequence

$$(1, 2, 4), (4, 8, 7), (12, 4, 1), (2, 12, 3), (4, 1, 6).$$

It is easy to see that $(4, 1, 6)$ is not obtainable from $(1, 2, 4)$ by one modular multiplication. However, our algorithm shows that

$$\begin{align*}
(1, 4, 9) & \rightarrow (4, 1, 6). \\
(5, 4, 15) & \rightarrow (1, 4, 5).
\end{align*}$$

Similarly, by applying pairs $(5, 8), (5, 12)$ we get the sequence

$$(1, 2, 4), (5, 2, 4), (1, 10, 8),$$

but now our algorithm shows that $(1, 10, 8)$ is not super-reachable at all.

The algorithm can be applied for other problems as well, e.g. for finding the smallest possible modulus for $B$. Then the complexity will increase; we have no explicit estimates.

Acknowledgment

The author has benefited very much from the discussions with and ideas of Tero Harju.
References