Acta Cybernetica **14** (1999) 27-36.

On ± 1-representations of integers

János Demetrovics*, Attila Pethő†and Lajos Rónyai*

This paper is dedicated to Professor Ferenc Gécseg on the occasion of his 60th birthday.

1 Introduction

In public key cryptography cryptosystems employing elliptic curves are playing an important role. Such systems are based on the elliptic version of the discrete logarithm problem. Let $\mathbb{K}$ be a finite field, $E = E(\mathbb{K})$ be an elliptic curve over $\mathbb{K}$ and let $P \in E$. If the binary expansion of $n \in \mathbb{N}$ is

$$n = \sum_{i=0}^{l} b_i 2^i, \quad b_i = 0, 1;\ i = 1, \ldots, l;\ b_l = 1,$$

then one can compute $P(n) = nP$ by using the following algorithm:

1. $P(n) \leftarrow P$
2. for $i \leftarrow 1$ to l do
   $\{P(n) \leftarrow 2P(n)$,
   if $b_{l-i} = 1$ then $P(n) \leftarrow P(n) + P.\}$

This algorithm requires l doubling and $\sum_{i=0}^{l-1} b_i$ addition steps. All operations are performed of course on the curve E. The idea is quite old. In a recipe for integer multiplication it appears in the Egyptian Rhind Papyrus dated from about 1650 B.C.

Observing that addition and subtraction on an elliptic curve have the same complexity, Morain and Olivos [MO] developed another algorithm for the computation of $P(n)$. Their algorithm uses one of the representations

$$n = \sum_{i=0}^{l'} d_i 2^i, \quad d_i = -1, 0, 1;\ i = 1, \ldots, l';\ d_{l'} = 1, \tag{1}$$

*Computer and Automation Institute, Hungarian Academy of Sciences, Budapest. The support of OTKA grants 016503, 016526, EC Grant ALTEC-KIT and FKFP grants 0612/1997, 0206/1997 is gratefully acknowledged.

†Institute of Mathematics, Kossuth Lajos University, Debrecen. Research supported in part by the Hungarian Foundation for Scientific Research, Grant N0. 25157/98, and by FKFP grant 0612/1997.

which we shall call a ± 1-representation of n. In the above algorithm only the conditional statement should be changed to

if $d_{l'-i} \neq 0$ then $P(n) \leftarrow P(n) + d_{l'-i}P$.

The new algorithm requires l' doubling and $\sum_{i=0}^{l'-1} |d_i|$ addition/subtraction steps on the curve. As $\sum_{i=0}^{l'-1} |d_i|$ can be considerably smaller than $\sum_{i=0}^{l-1} b_i$, the algorithm of Morain and Olivos may be more efficient, if l' is not too big compared to l.

We will point out to another application of ± 1-representation of integers. Let $A = (a_{ij})_{i=1,2;j=1,2}$ be a matrix with entries from a commutative ring, and with determinant ± 1. As $A^{-1} = \det(A)A^T$ the computation of A^{-1} means in this case only the swapping of $a_{1,2}$ and $a_{2,1}$, and the replacement of the sign of entries of A^T, whenever $\det(A) = -1$.

In contrast to the binary expansion, the ± 1-representation of integers is not at all unique. If, for example, the bitsequence of the binary expansion of n looks like $x01$, then $x0(-1)^k1$ are ± 1-representations of n for all $k \geq 0$. (Here we listed the digits in reverse compared to the usual representation.) Morain and Olivos [MO] (see also Müller [M]) describes two substitutions: $1^k \rightarrow -10^{k-1}1, k \geq 2$ and $1^k01^l \rightarrow -10^{k-1}1-10^{l-1}1 \rightarrow -10^{k-1}-10^l1$, which result usually ± 1-representation of smaller weight than the binary expansion. Moreover, both algorithms are linear in $\log n$. On the other hand, the length of the representation can be at most one longer than the shortest representations. For example the numbers $0^k11, k \geq 0$ have weight 2 and length $k+2$, but the algorithms of Morain and Olivos results $0^k - 101$, which has weight 2, but length $k+3$.

We call a ± 1-representation *optimal*, if $l' + \sum_{i=0}^{l'} |d_i|$ is minimal among the ± 1-representations of n. Note that the quantity $l' + \sum_{i=0}^{l'} |d_i|$ is actually one more than the number of additions/subtractions in E required when using the ± 1-representation (1) for computing nP. The aim of the present paper is to prove the following theorem.

Theorem 1 *There exists an algorithm which computes an optimal ± 1-representation of the integer n in $O(\log |n|)$ additions and comparisons.*

The proof of the theorem is constructive, i.e. we present a linear time algorithm for the computation of an optimal ± 1-representation of integers. Our method is the following: first we associate to the integer n an infinite, bipartite, directed acyclic graph $G(n)$ such that the ± 1-representations of n correspond to suitable directed paths in $G(n)$. Next we establish that to find an optimal ± 1-representation it suffices to consider a subgraph of $G(n)$ having at most $2\log_2 n + 5$ nodes. Our problem is actually equivalent to a single source shortest paths problem in this graph, which can be solved fast using a variant of the well known Dijkstra algorithm [D], [CLR].

2 The construction and elementary properties of $G(n)$

Let $0 \neq n \in \mathbb{N}$ and assume that 2^ν is the highest power of 2, which divides n. For each $k \geq 0$ we consider the solutions x of the congruence

$$x \equiv n \pmod{2^k}, \quad -2^k < x < 2^k. \tag{2}$$

This congruence has one solution, $x = n_k = 0$, if $0 \leq k \leq \nu$, and two solutions, if $k > \nu$. In the latter case we denote the solutions by $n_{k,1}$ and $n_{k,2}$ and order them as follows:

$$0 < |n_{k,2}| \leq 2^{k-1} \leq |n_{k,1}| < 2^k. \tag{3}$$

If $|n_{k,1}| = |n_{k,2}| > 0$ (which may happen only if $k = \nu + 1$), then put $n_{k,1} = 2^\nu$ and $n_{k,2} = -2^\nu$. The set of vertices V of $G(n)$ is

$$V = \{(k, n_k) : 0 \leq k \leq \nu\} \cup \{(k, n_{k,1}), (k, n_{k,2}) : k > \nu\}.$$

To lighten notation we shall refer to vertices (k, n_k) simply as n_k and $(k, n_{k,j})$ as $n_{k,j}$. Thus, in the sequel we will use the notations n_k and $n_{k,j}, j = 1, 2$ in two meanings; either as vertices of $G(n)$ or solutions of (2) satisfying the inequalities (3).

The set of edges E of $G(n)$ is the union of three sets, E_1, E_2, E_3, where

$$\begin{aligned} E_3 &= \{e_{k,1} = e_k = (n_k, n_{k+1}) : k = 0, \ldots \nu - 1\}, \\ E_2 &= \{e_{\nu,1} = (n_\nu, n_{\nu+1,1}), e_{\nu,2} = (n_\nu, n_{\nu+1,2})\}, \\ E_1 &= \{e_{k,j,h} = (n_{k,j}, n_{k+1,h}) : n_{k+1,h} = n_{k,j} + \varepsilon_{k,j,h} 2^k \text{ with} \\ &\quad \varepsilon_{k,j,h} \in \{-1, 0, 1\},\ k > \nu\}. \end{aligned}$$

Here $e = (x, y)$ means that the directed edge e joins vertex x to vertex y.

Let $d^-(x)$, (resp. $d^+(x)$) denote the indegree (the outdegree, resp.) of vertex $x \in V$, i.e. the number of edges having x as their endpoint (starting point, resp.). Now we prove the following simple lemma.

Lemma 1 *We have* $d^-(n_k) = d^-(n_{\nu+1,j}) = 1$, *if* $0 < k < \nu + 1, j = 1, 2$ *and* $d^-(n_{k,j}) = j$, *if* $k > \nu + 1, j = 1, 2$.

Proof: The first assertion is obvious. We consider therefore the second one. Let $k > \nu + 1$. An edge ending at vertex $n_{k,j}$, has, by construction, $n_{k-1,1}$ or $n_{k-1,2}$ as its starting point, and hence belongs to the set E_1. Then there exists an $h \in \{1, 2\}$ and $\varepsilon_{k-1,h,j} \in \{-1, 0, 1\}$ such that

$$n_{k,j} = n_{k-1,h} + \varepsilon_{k-1,h,j} 2^{k-1}.$$

Let first $j = 1$. If $\varepsilon = 0$ or $-sg(n_{k,1})$[1] then

$$|n_{k,1} - \varepsilon 2^{k-1}| \geq |n_{k,1}| \geq 2^{k-1} > |n_{k-1.h}|,\ h = 1, 2$$

[1] We denote by $sg(n)$ the sign of the integer n.

by (3). Hence there can be no edges, which correspond to these values of ε, i.e. $d^-(n_{k,1}) \leq 1$. On the other hand, if $\varepsilon = sg(n_{k,1})$ then for $u = n_{k,1} - sg(n_{k,1})2^{k-1}$ we have

$$|u| = |n_{k,1} - sg(n_{k,1})2^{k-1}| < 2^k - 2^{k-1} = 2^{k-1}.$$

This implies that $u = n_{k-1,1}$ or $u = n_{k-1,2}$ and hence $d^-(n_{k,1}) \geq 1$.

Let now be $j = 2$. If $\varepsilon = 0$ or $sg(n_{k,2})$ then

$$|n_{k,2} - \varepsilon 2^{k-1}| \leq |n_{k,2}| < 2^{k-1}$$

by (3). Hence there is one edge either from $n_{k-1,1}$ or from $n_{k-1,2}$ to $n_{k,2}$. If $\varepsilon = -sg(n_{k,2})$ then, as

$$|n_{k,2} - \varepsilon 2^{k-1}| > 2^{k-1},$$

by (3), there is no edge, which corresponds to this value of ε. Thus $d^-(n_{k,2}) = 2$, as asserted. □

Now we associate weights to the edges of $G(n)$. Let

$$w(e) = \begin{cases} 0, & \text{if } e \in E_3, \\ sg(n_{\nu+1,j}), & \text{if } e = (n_\nu, n_{\nu+1,j}) \in E_2, \\ sg(\varepsilon_{k,j,h}), & \text{if } e = (n_{k,j}, n_{k+1,h}) \in E_1. \end{cases}$$

The following lemma shows that the network $G(n)$ has a quite transparent structure.

Lemma 2 *If $k > \nu$, then there exists for every $\varepsilon \in \{-1, 0, 1\}$ exactly one pair of indices $(j, h), 1 \leq j, h \leq 2$ such that $w(e_{k,j,h}) = \varepsilon$. Moreover, for an edge $e_{k,j,h}$ we have $w(e_{k,j,h}) = 0$ if and only if $h = 2$ and $d^+(n_{k,j}) = 1$.*

Remark 1 *The second assertion of Lemma 2 means that if $k > \nu$ then the subgraph of $G(n)$ spanned by the nodes on the k-th and $k+1$-th levels has one of the following two types:*

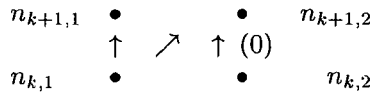


Figure 1

or

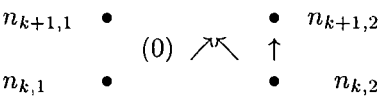


Figure 2

This observation will be important in the computation of an optimal ±1-representation of n.

Proof: We have seen in the proof of Lemma 1 that $w(e_{k,j,2}), j = 1, 2$ takes two different values: 0 and $sg(n_{k+1,2})$, while $w(e_{k,j,1}) = sg(n_{k+1,1})$. But $sg(n_{k+1,2}) \neq sg(n_{k+1,1})$, and both of them are different from 0, implying the first assertion.

¿From what we established so far, we know that $w(e_{k,j,h}) = 0$ implies that $h = 2$. It also follows from Lemma 1 that at level k one of the vertices has outdegree 1, the other has outdegree 2. Having all these, tò prove the second assertion, it suffices to verify that $w(e_{k,j,2}) = 0$ implies that $d^+(n_{k,j}) = 1$. The condition on the edge-weight gives that $n_{k+1,2} = n_{k,j}$. Using this we have

$$|n_{k+1,1} - n_{k,j}| = |n_{k+1,1} - n_{k+1,2}| = 2^{k+1} > \varepsilon 2^k,$$

for any $\varepsilon \in \{1, 0, -1\}$. This means that there can be no edge from $n_{k,j}$ to $n_{k+1,1}$, hence $d^+(n_{k,j}) = 1$. □

We have constructed an, in one direction, infinite directed acyclic graph $G(n)$. Observe, that if $2^k > |n|$, then $n_{k+j,2} = n$ for all $j \geq 1$. We shall prove, that this network describes completely the ±1-representations of the integer n.

To be more precise, let $U(n)$ denote the set of directed paths from 0 to the nodes n_{k,j_k} where $n_{k,j_k} = n$, and let

$$W(n) = \{(w(e_{0,j_1}), \ldots, w(e_{\nu,j_{\nu+1}}), w(e_{\nu+1,j_{\nu+1},j_{\nu+2}}), \ldots, w(e_{k-1,j_{k-1},j_k})),$$
$$\text{where the path } e_{0,j_1}, \ldots, e_{\nu,j_{\nu+1}}, e_{\nu+1,j_{\nu+1},j_{\nu+2}}, \ldots, e_{k-1,j_{k-1},j_k} \in U(n)\}.$$

Remark that $j_i = 1$, if $i \leq \nu$ and $j_i = 1$ or 2, otherwise. Hence $W(n)$ is the set of sequences of weights of directed path from the vertex $n_0 = 0$ to the vertices $n_{k,j_k} = n$.

On the other hand, let

$$E(n) = \{(d_0, \ldots, d_{k-1}), \text{ such that } n = \sum_{i=0}^{k-1} d_i 2^i,\ d_i \in \{-1, 0, 1\}\},$$

i.e. $E(n)$ is the set of sequences of digits of the ±1-representations of n. Now we are in the position to prove the following theorem.

Theorem 2 *If $n \neq 0$, then $W(n) = E(n)$.*

Proof: If $n < 0$ then we have obviously $W(-n) = -W(n)$ and $E(-n) = -E(n)$. Hence it is enough to prove the theorem for $n > 0$, which we assume in the sequel.

Let first

$$s = (w(e_{0,j_1}), \ldots, w(e_{\nu,j_{\nu+1}}), w(e_{\nu+1,j_{\nu+1},j_{\nu+2}}), \ldots, w(e_{k-1,j_{k-1},j_k})) \in W(n).$$

Then

$$\begin{aligned} & e_{0,j_1}, \ldots, e_{\nu,j_{\nu+1}}, e_{\nu+1,j_{\nu+1},j_{\nu+2}}, \ldots, e_{k-1,j_{k-1},j_k} \\ = \ & (n_0, n_1), \ldots, (n_{\nu-1}, n_\nu), (n_\nu, n_{\nu+1,j_{\nu+1}}), \\ & (n_{\nu+1,j_{\nu+1}}, n_{\nu+2,j_{\nu+2}}), \ldots, (n_{k-1,j_{k-1}}, n_{k,j_k}) \in U(n). \end{aligned}$$

We have the relations

$$n_{h,j_h} = n_{h-1,j_{h-1}} + w(e_{h-1,j_{h-1},j_h})2^{h-1},$$

whenever $h > \nu$, by the definition of the vertices and the weights. Hence, as $n_{k,j_k} = n$, we have

$$n = n_{k,j_k} = w(e_{\nu,j_{\nu+1}})2^{\nu} + \sum_{h=\nu+1}^{k-1} w(e_{h,j_h,j_{h+1}})2^h.$$

As $w(e_{h,h+1}) = 0$ for $h = 0, \ldots, \nu - 1$ we obtain

$$n = n_{k,j_k} = \sum_{h=0}^{\nu-1} w(e_{h,h+1})2^h + w(e_{\nu,j_{\nu+1}})2^{\nu} + \sum_{h=\nu+1}^{k-1} w(e_{h,j_h,j_{h+1}})2^h,$$

i.e. $s \in E(n)$.

Assume now that $s = (d_0, \ldots, d_{k-1}) \in E(n)$. Then

$$n = \sum_{i=0}^{k-1} d_i 2^i.$$

Let $n_0 = 0$, and if $0 < h \leq k$, then $n_h = \sum_{i=0}^{h-1} d_i 2^i$. Then $n_k = n$,

$$n_h \equiv n \pmod{2^h}$$

and

$$|n_h| \leq \sum_{i=0}^{h-1} 2^i < 2^h.$$

Thus, if $h > \nu$, then $n_h = n_{h,j_h}$ for $j_h = 1$ or $j_h = 2$. If $h < k$, then

$$n_{h+1} = n_h + d_h 2^h,$$

i.e

$$n_{h+1,j_{h+1}} = n_{h,j_h} + d_h 2^h.$$

This means that there exists an edge from n_{h,j_h} to $n_{h+1,j_{h+1}}$ and its weight is $w(e_{h,j_h,j_{h+1}}) = d_h$. Hence

$$(n_0, n_1), \ldots, (n_{\nu-1}, n_\nu), (n_\nu, n_{\nu+1,j_{\nu+1}}), (n_{\nu+1,j_{\nu+1}}, n_{\nu+2,j_{\nu+2}}), \ldots, (n_{k-1,j_{k-1}}, n_{k,j_k})$$

is a directed path from $n_0 = 0$ to $n_{k,j_k} = n$, i.e. $s \in W(n)$. The proof is complete. □

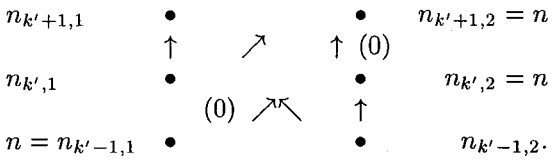
We define the *weight* of a node $n_{k,j}$ of $G(n)$ to be the minimum of the sums $\sum_{j=1}^{k} |w(e^j)|$, where the edges $e^1, e^2, \ldots, e^k$ form a directed path from 0 to $n_{k,j}$. We denote the weight of $n_{k,j}$ by $w(n_{k,j})$. In view of Theorem 2, our task of finding

an optimal representation of n is equivalent to finding a node $n_{k,j}$ with $k + w(n_{k,j})$ minimal among the nodes with $n_{k,j} = n$. (Please note that for an optimal node $n_{k,j}$ a shortest path from 0 to $n_{k,j}$ can not end with an edge of weight zero, hence for the ±1-representation of type (1) obtained from the path we have $l' = k$.)

To solve the latter optimization problem, we examine closely the lowest level of $G(n)$ where n appears as a vertex. Let $k' = \lfloor \log n \rfloor + 2$. Then $2^{k'-2} \leq n < 2^{k'-1}$. Hence $n_{k'-1,1} = n_{h,2} = n$ for all $h \geq k'$, by (3). Moreover, only these vertices are equal to n.

Lemma 3 *If $w(n_{k'-1,1}) \leq w(n_{k'-1,2}) + 1$ then (a shortest path from 0 to) node $n_{k'-1,1}$ provides an optimal representation of n. If $w(n_{k'-1,1}) > w(n_{k'-1,2}) + 1$ then (a shortest path to) node $n_{k',2}$ provides an optimal representation of n.*

Proof: By Remark 1. the layer of $G(n)$ comprising levels $k'-1, k'$ and $k'+1$ looks like



Here the edges without label have weight ±1. Using the fact that a directed path from 0 to a node at level $h \geq k'$ must pass through level $k'-1$, we have

$$w(n_{h,j}) \geq \min\{w(n_{k'-1,1}), w(n_{k'-1,2}) + 1\} = w(n_{k',2}).$$

Hence if $h > k'$ then $h + w(n_{h,j}) > k' + w(n_{k',2})$. ¿From this we see that the optimum is attained at node $n_{k'-1,1}$ or $n_{k,2}$.

If $w(n_{k'-1,1}) \leq w(n_{k'-1,2}) + 1$ then

$$k' - 1 + w(n_{k'-1,1}) \leq k' - 1 + w(n_{k',2}) < k' + w(n_{k',2}),$$

hence $n_{k'-1,1}$ is the (only) optimal node.

On the other hand, if $w(n_{k'-1,1}) > w(n_{k'-1,2}) + 1$ then $w(n_{k'-1,1}) > w(n_{k',2})$, and therefore $k' - 1 + w(n_{k'-1,1}) \geq k' + w(n_{k',2})$. In this case $n_{k',2}$ is an optimal node. □

Note that the lemma implies in particular that the length l' of an optimal ±1-representation (1) of n can have at most two values. The second alternative of the Lemma 3 allows for the possibility of two optimal nodes. This may indeed happen, as exemplified by the representations $7 = 4 + 2 + 1$ and $7 = 8 - 1$.

Proof of Theorem 1. The algorithm now is quite straightforward to outline. On input $n > 0$ we build the the first k' layers of the graph $G(n)$ and calculate the the edge weights. It is a directed acyclic graph (dag) with no more than $2\log_2 n + 5$ vertices and $3\log_2 n + 6$ edges. Following the definition directly, this graph can be built using $O(\log n)$ elementary operations.

By Lemma 3 it suffices to compute the weights $w(n_{k'-1,1})$, $w(n_{k',2})$ together with with an appropriate path from 0 to $n_{k'-1,1}$ or to $n_{k',2}$ which provides the optimal weight. We can use here Dijkstra's algorithm for the single source shortest path problem. In doing this, we have to work with the absolute values of the original edge-weights. Dijkstra's method can be implemented in linear time for dag-s (see for example section 25.4 in [CLR]), hence this phase can also be accomplished in time $O(\log n)$. □

3 A detailed algorithm

Here we present a detailed and streamlined procedure which performs the tasks sketched in the proof of Theorem 1. It computes an optimal ± 1-representation of the input integer $n > 0$. In the following description we use a two-dimensional array $n(h,j), j = 1,2$, to represent the vertices $n_{h,j}$ of the network $G(n)$. The value of $n(h,j)$ is a three-dimensional vector, whose i-th coordinate will be denoted by $n(h,j)[i]$.

Upon termination $n(h,j)[3]$ will store $w(n_{h,j})$. Moreover, $n(h,j)[1]$ stores an identifier of the next to last vertex of an optimal path to $n_{h,j}$, and $n(h,j)[2]$ contains the weight of the last edge along this path. More formally, in the general situation (i.e. if $h > \nu$) we intend to achieve the following:

$$
\begin{aligned}
n(h,j)[3] &= \min\{n(h-1,\ell)[3] + |w(e_{h-1,j,\ell})|, \text{ where } e_{h-1,j,\ell} \in G(n)\},\\
n(h,j)[2] &= w(e_{h-1,j,\ell}), \text{ if } n(h,j)[3] = n(h-1,\ell)[3] + |w(e_{h-1,j,\ell})|,\\
n(h,j)[1] &= \ell, \text{ if } n(h,j)[3] = n(h-1,\ell)[3] + |w(e_{h-1,j,\ell})|.
\end{aligned}
$$

Algorithm

Input: $n > 0$ an integer
Output: $(d_0, \ldots, d_{k-1})$ an optimal ± 1-representation of n.

1. $k' := \lfloor \log n \rfloor + 2$
2. Compute $G(n)$ up to level k'
3. **for** $h := 1$ **to** ν **do** $n(h,1) := (1,0,0)$
4. $n(\nu+1,1) := (1, w(e_{\nu,\nu+1,1}), 1); n(\nu+1,2) := (1, w(e_{\nu,\nu+1,2}), 1)$
5. **for** $h := \nu + 2$ **to** k' **do**
   if $e_{h-1,1,1} \in G(n)$ **then begin**
   $n(h,1) := (1, w(e_{h-1,1,1}), n(h-1,1)[3] + 1)$
   $n(h,2) := (2, 0, n(h-1,2)[3])$
   if $n(h-1,1)[3] + 1 < n(h,2)[3]$ **then**
   $n(h,2) := (1, w(e_{h-1,1,2}), n(h,1)[3])$
   end

else begin
$n(h,1) := (2, w(e_{h-1,2,1}), n(h-1,2)[3]+1)$
$n(h,2) := (1, 0, n(h-1,1)[3])$
if $n(h-1,2)[3]+1 < n(h,2)[3]$ **then**
$n(h,2) := (2, w(e_{h-1,2,2}), n(h,1)[3])$
end

6. $k := k'-1; j := n(k,1)[1]; d := (n(k,1)[2])$
if $n(k',2)[3] < n(k,1)[3]$ **then** $k := k'; j := n(k,2)[1]; d := (n(k,2)[2])$

7. **while** $k \neq 0$ **do**
d:=(n(k,j)[2],d); j:=n(k,j)[1]; k:=k-1

8. **output** d.

Proposition 1 *The preceding Algorithm computes an optimal ±1-representation of the integer n in $O(\log n)$ time.*

Proof: It is clear that the algorithm terminates after $O(\log n)$ steps. Therefore, it is enough to establish correctness.

The basis of the calculation of $w(n_{h,j})$ is the straightforward relation

$$w(n_{h,j}) = \min\{w(n_{h-1,j}) + |w(e_{h-1,i,j})|, \text{ where } \quad (n_{h-1,i}, n_{h,j}) \in G(n)\}.$$

As $w(e) = 0$ for $e \in E_3$, we have $w(n_{h,j}) = 0$ for $h \leq \nu$. Thus $n(h,1)[3]$ is set correctly in Step 3 for $1 \leq h \leq \nu$. The same is true for $n(\nu+1, j)[3], j = 1, 2$, because $|w(e)| = 1$ for $e \in E_2$. If $h \geq \nu + 2$ then the h-th level of $G(n)$ has one of the shapes, presented on Figures 1 and 2. The value of $n(h,j)[3]$ is determined in Step 5 according to these alternatives. Thus $n(h,j)[3] = w(n_{h,j})$ for all h and j considered. By Lemma 3 it is enough to compute the weights up until level $\lfloor \log n \rfloor + 2$, hence k' is set properly in Step 1.

Lemma 3 shows also that in Step 6 the parameters k, j of an optimal node $n_{k,j}$ are calculated correctly. In fact, we set $k = k' - 1$, if $w(n_{k'-1,1}) \leq w(n_{k',2})$, and $k = k'$, if $w(n_{k'-1,1}) > w(n_{k',2})$. Finally, by tracing backwards an optimal path to $n_{k,j}$ in loop 7, we compute the digits of an optimal ±1-representation. The proposition is proved. □

Acknowledgement We thank I. Ruzsa and S. Turjányi for their comments and helpful conversations during the preparation of this paper.

References

[CLR] T.H. CORMEN, C.E.LEISERSON and R.L. RIVEST, *Introduction to algorithms*, The MIT Press, 1990.

[D] E.W. DIJKSTRA, *A note on two problems in connexion with graphs*, Numer. Math. **1** (1959), 269-271.

[Me] A. MENEZES, *Elliptic curve public key cryptosystems*, Kluwer Academic Publishers, 1993.

[M] V. MÜLLER, *Efficient algorithms for multiplication on elliptic curves* to appear.

[MO] F. MORAIN and J. OLIVOS, *Speeding up the computations on an elliptic curve using addition-subtraction chains*, in F. MORAIN, *Courbes elliptiques et tests de primalité*, Doctoral Thesis, Université Lyon I, 1990.