Cryptogr. Commun. (2016) 8:513–523
DOI 10.1007/s12095-015-0157-2

Determining the k-error joint linear complexity spectrum for a binary multisequence with period p^n

Fulin Li[1] · Shixin Zhu[1] · Honggang Hu[2] · Ting Jiang[1]

Received: 9 July 2014 / Accepted: 18 August 2015 / Published online: 7 September 2015

Abstract Recent developments in word-based stream ciphers present the study on multisequences. The joint linear complexity and k-error joint linear complexity are fundamental concepts for the assessment of multisequences. The k-error joint linear complexity spectrum contains all the information about how the joint linear complexity of a multisequence decreases as the number k of allowed bit changes increases. In this paper, we present an efficient algorithm by which the k-error joint linear complexity spectrum for a t-fold p^n-periodic binary multisequence can be entirely determined using $\mathcal{O}(tp^n \log p)$ bit operations, where p is an odd prime, 2 is a primitive root modulo p^2 and n is a positive integer.

Keywords Stream ciphers · Multisequence · Algorithm · Error joint linear complexity spectrum

This research is supported by National Natural Science Foundation of China (No.61271271 and 61370089); Natural Science Foundation of Anhui Province under Grant 1508085MA13; 100 Talents Program of Chinese Academy of Science and the Fundamental Research Funds for the Central Universities (J2014HGXJ0075).

✉ Fulin Li
lflsxx66@163.com

Shixin Zhu
zhushixin@hfut.edu.cn

Honggang Hu
hghu2005@ustc.edu.cn

Ting Jiang
ahjiangting@126.com

[1] Department of Applied Mathematics, Hefei University of Technology, Hefei 230009, Anhui, People's Republic of China

[2] School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, Anhui, People's Republic of China

Mathematics Subject Classification (2010) 94A55 · 94A60

1 Introduction

The linear complexity and the k-error linear complexity of periodic sequences over finite fields are important randomness measures for pseudorandom sequence generators employed in stream ciphers. The linear complexity of a sequence is the length of the shortest linear feedback shift register (LFSR) which generates the sequence. If the shortest LFSR has length of l, then $2l$ consecutive bits of the output can determine its feedback polynomial in terms of Berlekamp-Massey algorithm [1, 2]. In other words, if the linear complexity of a sequence is l, then the Berlekamp-Massey algorithm requires just $2l$ bits to recover the whole sequence. Therefore, to resist known-plaintext attack, the linear complexity of a sequence must be enough large. But the sequence with large linear complexity may not be difficult to predict. If the linear complexity of a sequence immediately decreases as few bits in one period are changed, then the sequence is still cryptographically weak because we can at least determine a sequence which coincides with the correct sequence in all but a "small" number of positions. Thus the linear complexity of a sequence should also remain large even if some of its terms are changed. This observation leads to the study of the k-error linear complexity. The k-error linear complexity of a sequence s with period N is the minimal linear complexity that can be obtained for s by modifying up to k terms in one period (and modifying all other periods in the same way). This notion was defined in [3] and is closely related to previously defined notions of sphere complexity [4] and weight complexity [5]. In [6], Niederreiter showed that there is a class of periodic sequences which possesses large linear complexity and large k-error linear complexity simultaneously.

There exist several efficient algorithms [3, 7–9] which compute the linear complexity and k-error linear complexity of a given periodic sequence. Stamp and Martin [3] extended the Games-Chan algorithm [7] to compute the k-error linear complexity of a binary sequence of period $l = 2^n$ using $\mathcal{O}(l \log l)$ bit operations. The Stamp and Martin algorithm can be used to compute entirely linear complexity spectrum of a period $l = 2^n$ sequence using $\mathcal{O}(l^2 \log l)$ bit operations. Further, Lauder and Paterson [10] showed that the whole error linear complexity spectrum (i.e., the k-error linear complexity for each value of k) of a binary sequence of period $l = 2^n$ can be computed using $\mathcal{O}(l(\log l)^2)$ bit operations. The k-error linear complexity spectrum contains all the information about how the linear complexity of a sequence decreases as the number k of allowed bit changes increases. However, it remains a challenging open problem to devise an algorithm which efficiently computes the error linear complexity spectrum of a binary sequence of general period. Recent developments in stream ciphers point towards an interest in word-based (or vectorized) stream ciphers [11–17]. The theory of such stream ciphers requires the study of the complexity measures for multisequences, i.e., for parallel streams of finite many sequences. In this direction, joint linear complexity and k-error joint linear complexity of multisequences have been investigated [13–17]. Let $\mathbb{F}_2$ be the finite field with 0,1 elements, and let t and N be positive integers. An t-fold N-periodic multisequences $\mathbf{S}$ over $\mathbb{F}_2$ is of the form $\mathbf{S} = (S[1], S[2], \cdots, S[t])$, where $S[\lambda] = (S[\lambda, 0], S[\lambda, 1], \cdots, S[\lambda, N-1])^\infty$ is an N-periodic sequence with bits in $\mathbb{F}_2$ for each $\lambda = 1, 2, \cdots, t$.Actually, $(S[\lambda, 0], S[\lambda, 1], \cdots, S[\lambda, N-1])^\infty$ denotes constantly duplication of successive N terms in the firstly period of sequence $S[\lambda]$. By N-periodic we note that N is a period length of the sequence, but not necessarily the least period length. The joint linear complexity $LC(\mathbf{S})$ is the least order of a linear recurrence relation that $S[1], S[2], \cdots, S[t]$ satisfy simultaneously, with $LC(\mathbf{S}) = 0$ if $\mathbf{S}$ is the zero multisequence. For multisequences, the definition of k-error joint linear complexity was presented in [14–16]. For an integer k with $0 \le k \le tN$, the k-error joint linear complexity of t-fold N-periodic multisequence is defined by $LC_k(\mathbf{S}) = \min_{\mathbf{T}} LC(\mathbf{T})$, where the minimum is taken over all N-periodic t-fold multisequences $\mathbf{T}$ with term distance $d(\mathbf{S}, \mathbf{T}) \le k$. The term distance $d(\mathbf{S}, \mathbf{T})$ is defined to be the number of terms in $\mathbf{S}$ that are different from the corresponding terms in $\mathbf{T}$. Especially, for $k = 0$, k-error joint linear complexity is its joint linear complexity for a multisequence. The paper [15] marks the beginning of the theory of k-error joint linear complexity measures for multisequences. As suggested in that paper, an interesting task would be to find analogs of all major results on the k-error linear complexity of single sequences for the case of multisequences. Sethumadhavana et al. presented an algorithm (see in [17]) to compute the k-error joint linear complexity of 2^n-periodic multisequence over $\mathbb{F}_2$. We presented an algorithm (See [16]) to compute the k-error joint linear complexity of a p^n-periodic multisequence over $\mathbb{F}_q$, where p is an odd prime, q a primitive root modulo p^2. The k-error joint linear complexity spectrum contains all the information about how the joint linear complexity of a multisequence decreases as the number k of allowed bit changes increases.

In this paper, we present an efficient algorithm by which the k-error joint linear complexity spectrum for a p^n-periodic binary multisequence can be entirely determined, where p is an odd prime, 2 a primitive root modulo p^2 and n a positive integer. On one hand, our main results in Section 3, when reduced to the case of single sequence, i.e., if $t = 1$, are actually further generalization of results in [10]; On the other hand, although the algorithm [16] can also be used to determine entirely the k-error joint linear complexity spectrum of a p^n-periodic binary multisequence, it requires $\mathcal{O}(t(p-1)p^n \log p^n)$ bit operations, where p is an odd prime and 2 a primitive root modulo p^2. This paper's contribution is to accomplish the same task, i.e., compute entirely the k-error joint linear complexity spectrum, using $\mathcal{O}(tp^n \log p)$ bit operations by presenting an algorithm.

This paper is organized as follows. In Section 2, we give necessary definitions and present some preliminary results, and explain the concept of cost binary multisequences. Section 3 contains pseudo-code for the main algorithm of the paper, as well as a proof of the correctness and an analysis of the computational complexity of this algorithm, which compute the k-error joint linear complexity spectrum of a binary multisequence $\mathbf{S}$ by computing the k-error joint linear complexity spectrum of its costed binary multisequence $\mathcal{S} = (\mathbf{S}, \sigma, l)$. Section 4 concludes this paper by presenting an interesting open problem.

2 Definitions and basic results

Let $\mathbf{S} = (S[1], S[2], \cdots, S[t])$ be a binary multisequence with length l, where $S[\lambda] = (s[\lambda, 0], s[\lambda, 1], \cdots, s[\lambda, l-1])$, $1 \le \lambda \le t$ denote the λth single sequence of $\mathbf{S}$. Let $\sigma = (\sigma[1], \sigma[2], \cdots, \sigma[t])$ is a nonnegative real multisequence with length l, where $\sigma[\lambda] = (\sigma[\lambda, 0], \sigma[\lambda, 1], \cdots, \sigma[\lambda, l-1])$, $1 \le \lambda \le t$; and l is a positive integer. Let triple $\mathcal{S} = (\mathbf{S}, \sigma, l)$. We call the triple $\mathcal{S} = (\mathbf{S}, \sigma, l)$ a costed binary multisequence and σ the cost multisequence of $\mathbf{S}$. In this paper, our main algorithm is carried out step by step. In fact, $\sigma[\lambda, j]$ is intended to measure the cost of changing the current one bit $s[\lambda, j]$, $0 \le j \le l-1$ without disturbing the results of any previous step. The cost is the contribution to error.

Especially, when $\sigma[\lambda, i] = 1$ for all λ and i, $1 \leq \lambda \leq t, 0 \leq i \leq l-1$, the cost is the Hamming weight of error vector. The notation is similar to the cost of single sequence which is introduced firstly in [1].

Now, we suppose that the period of a binary multisequence is $l = p^n$, where p is an odd prime, 2 a primitive root modulo p^2 and n a positive integer. In fact, $l = p^n$ can be comprehended to be initial value of variable l. We define the two maps $B(\mathcal{S}) = (B(\mathbf{S}), B(\sigma), \frac{l}{p})$ and $D(\mathcal{S}) = (D(\mathbf{S}), D(\sigma), \frac{l}{p})$.

The pseudo-code definitions of $B(\mathcal{S}) = (B(\mathbf{S}), B(\sigma), \frac{l}{p})$ and $D(\mathcal{S}) = (D(\mathbf{S}), D(\sigma), \frac{l}{p})$ are as follows:

(1) The B map:

```
Input    $\mathcal{S} = (\mathbf{S}, \sigma, l)$
Output   $B(\mathcal{S}) = (B(\mathbf{S}), B(\sigma), \frac{l}{p})$
For      $0 \leq i < \frac{l}{p}$
              $B(\mathbf{S})[\lambda, i] = \sum_{j=0}^{p-1} s[\lambda, i + j\frac{l}{p}]$
              $B(\sigma)[\lambda, i] = \min\{\sigma[\lambda, i + j\frac{l}{p}] : j = 0, 1, \cdots, p-1\}$
```

(2) The D map:

```
Input    $\mathcal{S} = (\mathbf{S}, \sigma, l)$
Output   $D(\mathcal{S}) = (D(\mathbf{S}), D(\sigma), \frac{l}{p})$
For      $0 \leq i < \frac{l}{p}, 1 \leq \lambda \leq t$
              $T_{\lambda,i0} = \sum_{j=0}^{p-1} \sigma[\lambda, i + j\frac{l}{p}](s[\lambda, i + j\frac{l}{p}] \oplus 1)$
              $T_{\lambda,i1} = \sum_{j=0}^{p-1} \sigma[\lambda, i + j\frac{l}{p}](s[\lambda, i + j\frac{l}{p}])$
              $if\ s[\lambda, i] = s[\lambda, i + \frac{l}{p}] = \cdots = s[\lambda, i + (p-1)\frac{l}{p}]$
                $D(\mathbf{S})[\lambda, i] = s[\lambda, i]; D(\sigma)[\lambda, i] = \sum_{j=0}^{p-1} \sigma[\lambda, i + j\frac{l}{p}]$
              $else$
                $if\ T_{\lambda,i0} > T_{\lambda,i1}$
                  $D(\mathbf{S})[\lambda, i] = 0, D(\sigma)[\lambda, i] = T_{\lambda,i0} - T_{\lambda,i1}$
                $else$
                  $D(\mathbf{S})[\lambda, i] = 1, D(\sigma)[\lambda, i] = T_{\lambda,i1} - T_{\lambda,i0}$
```

Now, we can present two important results in [16] into the forms of Lemmas 1 and 2 as follows.

Lemma 1 [16] *Let* $\mathbf{S}$ *be a* t*-fold binary multisequence with period* p^n*, where* p *is an odd prime and 2 is a primitive root modulo* p^2*. Then*
$LC(\mathbf{S}) = LC(D(\mathbf{S}))$*, if* $s[\lambda, i] = s[\lambda, i + \frac{l}{p}] = \cdots = s[\lambda, i + (p-1)\frac{l}{p}]$, $1 \le \lambda \le t, 0 \le i \le \frac{l}{p} - 1$*; else,* $LC(\mathbf{S}) = (p-1)p^{n-1} + LC(B(\mathbf{S}))$.

Definition 2 Let $\mathcal{S} = (\mathbf{S}, \sigma, l)$ be a costed binary multisequence and $\mathbf{E} = (E[1], E[2], \cdots, E[t])$ be a binary multisequence with length l, where $E[\lambda] = (e[\lambda, 0], e[\lambda, 1], \cdots, e[\lambda, l-1])$, $1 \le \lambda \le t$. Set $cost(\mathbf{S} \to \mathbf{S} \oplus \mathbf{E}) = \sum_{e[\lambda,i]=1} \sigma[\lambda, i]$ and then the k-error joint linear complexity of the costed binary multisequence $\mathcal{S}$ is defined as

$$LC_k(\mathcal{S}) = \min_{cost(\mathbf{S} \to \mathbf{S} \oplus \mathbf{E}) \le k} LC(\mathbf{S} \oplus \mathbf{E})$$

where $\mathbf{E}$ is called the error multisequence of $\mathcal{S} = (\mathbf{S}, \sigma, l)$.

If $\sigma[\lambda, i] = 1$ for all λ and i, then the k-error joint linear complexity of $\mathcal{S}$ given here agrees with the k-error joint linear complexity $\mathbf{S}$ for a binary multisequence in [14, 15].

On the basis of the definition of two maps B and D, by Lemma 1, the algorithm 2 in [16] can be expressed in the following form.

Lemma 2 [16] *Let* $\mathcal{S} = (\mathbf{S}, \sigma, l)$ *be a costed binary multisequence, where* $l = p^n$*,* p *is an odd prime, and 2 is a primitive root modulo* p^2*. Let* $T_{\lambda,i0} = \sum_{j=0}^{p-1} \sigma[\lambda, i + j\frac{l}{p}](s[\lambda, i + j\frac{l}{p}] \oplus 1)$*, and* $T_{\lambda,i1} = \sum_{j=0}^{p-1} \sigma[\lambda, i + j\frac{l}{p}] s[\lambda, i + j\frac{l}{p}]$*,* $T = \sum_{\lambda=1}^{t} \sum_{i=0}^{l-1} \min\{T_{\lambda,i0}, T_{\lambda,i1}\}$*. Then the* k*-error joint linear complexity of multisequence* $\mathbf{S}$ *is as follows*

(1) $LC_k(\mathcal{S}) = (p-1)p^{n-1} + LC_k(B(\mathcal{S}))$*, for* $0 \le k < T$*;*

(2) $LC_k(\mathcal{S}) = LC_{k-T}(D(\mathcal{S}))$*, for* $k \ge T$*.*

Remark 1 Actually, readers can directly prove Lemma 2 by using the method of Lemma 3 in [10]. However, it is found that process of proof for multisequence may be extremely tedious without any technical improvements by using the method in [10]. Here, we present an alternative method to redisplay Lemma 2 on the basis of our results in [16].

Remark 2 By Lemma 1, the joint linear complexity does not increase if and only if $s[\lambda, i] = s[\lambda, i + \frac{l}{p}] = \cdots = s[\lambda, i + (p-1)\frac{l}{p}]$, for every λ and i, $1 \le \lambda \le t, 0 \le i \le l - 1$. If we add k allowable errors to a binary multisequence so as to force $s[\lambda, i] = s[\lambda, i + \frac{l}{p}] = \cdots = s[\lambda, i + (p-1)\frac{l}{p}]$, for every λ and i, $1 \le \lambda \le t, 0 \le i \le l - 1$, then the number of k bits is at least $T = \sum_{\lambda=1}^{t} \sum_{i=0}^{l-1} \min\{T_{\lambda,i0}, T_{\lambda,i1}\}$.

Definition 3 Let $\mathcal{S} = (\mathbf{S}, \sigma, l)$ be a costed binary multisequence. The k-error joint linear complexity spectrum (EJLCS) of $\mathcal{S}$ is defined as the k-error joint linear complexity sequence of $\mathcal{S}$:

$$LC_0(\mathcal{S}), LC_1(\mathcal{S}), \cdots, LC_{cost(\mathbf{S}\to\mathbf{0})}(\mathcal{S}).$$

The spectrum is actually composed by the set of the ordered list of $\{(k, LC_k(\mathcal{S})) : 0 \leq k \leq cost(\mathbf{S} \to \mathbf{0})\}$, where $cost(\mathbf{S} \to \mathbf{0})$ is actually $cost(\mathbf{S} \to \mathbf{S} \oplus \mathbf{S})$ on the basis of Definition 2.

Note that we do not restrict k to be an integer, so the EJLCS of $\mathcal{S}$ is not a finite set. However, it is clear from Definition 3 that $LC_k(\mathcal{S})$ takes on only finitely many values, so the EJLCS of $\mathcal{S}$ can be visualized as a graph with axes for cost and joint linear complexity and with points $(k, LC_k(\mathcal{S})), 0 \leq k \leq cost(\mathbf{S} \to 0)$.

Lemma 3 *Let $\mathcal{S} = (\mathbf{S}, \sigma, p^n)$ be a costed binary multisequence, where p is an odd prime and 2 a primitive root modulo p^2. Let $B(\mathcal{S}) = (B(\mathbf{S}), B(\sigma), p^{n-1})$ with $T = \sum_{\lambda=1}^{t} \sum_{i=0}^{p^{n-1}-1} \min\{T_{\lambda,i0}, T_{\lambda,i1}\}$ and $D(\mathcal{S}) = (D(\mathbf{S}), D(\sigma), p^{n-1})$ with $U = cost(D(\mathbf{S}) \to \mathbf{0})$. Then the EJLCS of multisequence $\mathcal{S}$ is*

$$\{(k, LC_k(B(\mathcal{S})) + (p-1)p^{n-1}) : 0 \leq k < T\} \bigcup \{(T + k, LC_k(D(\mathcal{S}))) : 0 \leq k \leq U\}.$$

Proof By Definition 3 of the EJLCS of multisequence $\mathcal{S}$, we immediately obtain the correctness of Lemma 3 from Lemma 2. □

Definition 4 Let $(k, LC_k(\mathcal{S}))$ be a point on the EJLCS of the costed binary multisequence $\mathcal{S}$. We say that $(k, LC_k(\mathcal{S}))$ is critical if for all points $(k^{'}, LC_{k^{'}}(\mathcal{S})$ of the EJLCS with $k^{'} < k$ we have that $LC_{k^{'}}(\mathcal{S}) > LC_k(\mathcal{S})$.

In others words, critical points are the points on the graph of the EJLCS where a decrease in the k-error joint linear complexity occurs. The sublist of all critical points in the EJLCS of $\mathcal{S}$ is called the critical error joint linear complexity spectrum (CEJLCS) of $\mathcal{S}$. We observe that the EJLCS of $\mathcal{S}$ contains the point $(0, LC(\mathcal{S}))$. Because the joint linear complexity of $\mathbf{S} \oplus \mathbf{E}$ can take on only finitely many different values, the CEJLCS of $\mathcal{S}$ contains a finite number of points.‘We observe that the EJLCS of $\mathcal{S}$ contains the point $(0, LC(\mathcal{S}))$.Note that the CEJLCS of a costed binary multisequence entirely determines its EJLCS and vice verse, and we use the terminology "critical" here. In Example 1 below text, the CEJLCS of the period 27 binary multisequence $\mathcal{S}$ is the set

$$\{(0, 27), (1, 26), (2, 25), (3, 24), (4, 8), (6, 7), (9, 6), (13, 2), (21, 1), (33, 0)\}.$$

Lemma 4 *Let $\mathcal{S} = (\mathbf{S}, \sigma, p^n)$ be a costed binary multisequence, where p is an odd prime and 2 a primitive root modulo p^2. Let $B(\mathcal{S}) = (B(\mathbf{S}), B(\sigma), p^{n-1})$ with $T = \sum_{\lambda=1}^{t} \sum_{i=0}^{p^{n-1}-1} \min\{T_{\lambda,i0}, T_{\lambda,i1}\}$ and $D(\mathcal{S}) = (D(\mathbf{S}), D(\sigma), p^{n-1})$ with $U = cost(D(\mathbf{S}) \to 0)$. Let the CEJLCS of $B(\mathcal{S})$ be $\{(k_i, LC_{k_i}(B(\mathcal{S}))) : 0 \leq i \leq m\}$ for some m, where $k_0 = 0$ and $k_m = T$, and the CEJLCS of $D(\mathcal{S})$ be*

$\{(K_i, LC_{K_i}(D(\mathcal{S}))) : 0 \leq i \leq u\}$ for some u, where $K_0 = 0$ and $K_u = U$. Then the CEJLCS of $\mathcal{S}$ is

$$\{(k_0, LC_{k_0}(B(\mathcal{S})) + (p-1)p^{n-1}), \cdots, (k_{m-1}, LC_{k_{m-1}}(B(\mathcal{S})) + (p-1)p^{n-1})\}$$

$$\bigcup \{(T + K_0, LC_{K_0}(D(\mathcal{S}))), \cdots, (T + K_u, LC_{K_u}(D(\mathcal{S})))\}.$$

Proof Note that the CEJLCS of $B(\mathcal{S})$ and $D(\mathcal{S})$ are their respective critical point list. By Definition 4, we can suppose all critical points of $B(\mathcal{S})$ are $(k_0, LC_{k_0}(B(\mathcal{S})) + (p-1)p^{n-1}), \cdots, (k_{m-1}, LC_{k_{m-1}}(B(\mathcal{S})) + (p-1)p^{n-1})$, where $k_i \leq T, i = 0, 1, \cdots, m-1$; and all critical points of $D(\mathcal{S})$ are $(K_0, LC_{K_0}(D(\mathcal{S}))), \cdots, (K_u, LC_{K_u}(D(\mathcal{S})))$, where $0 \leq K_i \leq U$. From Lemma 3, we obtain immediately the result. □

3 Computing CEJLCS of a costed binary multisequence

In this subsection we present an algorithm for computing the CEJLCS of a costed binary multisequence with period p^n, where p is an odd prime and 2 a primitive root modulo p^2. The algorithm is recursive, calling itself with progressively shorter costed multisequences as input. It can be thought of as exploring a binary tree where a node at depth i $(0 \leq i \leq n)$ corresponds to a costed binary multisequence of length p^{n-i} and the two edges emanating from a node correspond to the two mappings B and D that can be applied to this multisequence (Fig. 1).

As well as a costed binary multisequence $\mathcal{S} = (\mathbf{S}, \sigma, p^n)$, the algorithm has three integers tsf, lim, and c as inputs. We now explain briefly the three variable functions tsf, lim, and c . At any stage in the execution of the algorithm, the variable tsf is set to the total cost of

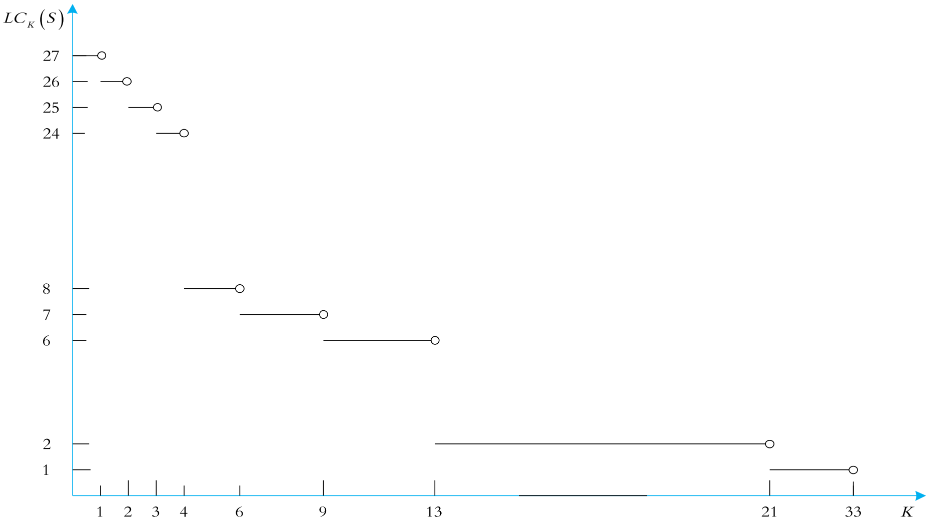

Fig. 1 The graph of the CEJLCS of multisequence $\mathcal{S} = (\mathbf{S}, \sigma, 27)$

changes made to the multisequence so far, and the variable lim denotes a limit to the total cost of changes one should consider when searching a particular part of the tree.

Algorithm 1 $CEJLCS(\mathcal{S} = (\mathbf{S}, \sigma, l), tsf, lim, c), l = p^n$

```
$if \quad l > 1$
    $computing B(\mathcal{S}), D(\mathcal{S}), T = \sum_{\lambda=1}^{t} \sum_{i=0}^{l-1} \min\{T_{\lambda,i0}, T_{\lambda,i1}\}$
  $if\ T > 0$
      $\{CEJLCS((B(\mathbf{S}), B(\sigma), \frac{l}{p}), tsf, \min\{lim, tsf + T - 1\}, c + \frac{(p-1)l}{p})\}$
  $if\ tsf + T \le lim$
      $\{CEJLCS((D(\mathbf{S}), D(\sigma), \frac{l}{p}), tsf + T, lim, c)\}$
  $else \backslash *l = 1 * \backslash$
      $if (s[1,0], s[2,0], \cdots, s[t,0]) = \vec{0}\ output\ (tsf, c)$
      $if (s[1,0], s[2,0], \cdots, s[t,0]) \ne \vec{0}\ and\ \sum_{\lambda=1}^{t} \sigma[\lambda, 0] > 0\ output\ (tsf, c+1)$
      $if (s[1,0], s[2,0], \cdots, s[t,0]) \ne \vec{0}\ and\ tsf + \sum_{\lambda=1}^{t} \sigma[\lambda, 0] \le lim\ output$
          $(tsf + \sum_{\lambda=1}^{t} \sigma[\lambda, 0], c)$
```

Theorem 1 *Algorithm CEJLCS* $((\mathbf{S}, \sigma, l), tsf, lim, 0))$*, where* $tsf \le lim$*, outputs the following list of points:*

$$\left\{(tsf + k_0, c + LC_{k_0}(\mathcal{S})), (tsf + k_1, c + LC_{k_1}(\mathcal{S})), \cdots, (tsf + k_u, c + LC_{k_u}(\mathcal{S}))\right\}$$

where $(k_0, LC_{k_0}(\mathcal{S})), \cdots, (k_u, LC_{k_u}(\mathcal{S}))$ *are the critical points of* $\mathcal{S} = (\mathbf{S}, \sigma, l)$ *whose first coordinates lie in the range* $[0, lim - tsf]$.

Proof Firstly, we prove the algorithm CEJLCS $((\mathbf{S}, \sigma, l), tsf, \text{lim}, c))$, where $tsf \le lim$, outputs the following list of points

$$\left\{(tsf + k_0, c + LC_{k_0}(\mathcal{S})), (tsf + k_1, c + LC_{k_1}(\mathcal{S})), \cdots, (tsf + k_u, c + LC_{k_u}(\mathcal{S}))\right\}$$

where $(k_i, LC_{k_i}(\mathcal{S})),\ k_i \in [0, lim - tsf],\ 0 \le i \le u$ are the critical points of $\mathcal{S} = (\mathbf{S}, \sigma, l)$. For $l = 1$, there exist two cases as follows

(1) $(s[1,0], s[2,0], \cdots, s[t,0]) = \vec{0}$, i.e., $cost(\mathbf{S} \to 0) = 0$. Note that there is the single critical point $(0, 0)$. The algorithm outputs (tsf, c) as required.

(2) $s[1,0], s[2,0], \cdots, s[t,0]) \ne \vec{0}$, i.e., $cost(\mathbf{S} \to 0) > 0$. If $\sum_{\lambda=1}^{t} \sigma[\lambda, 0] \le lim - tsf$, then there exist two critical points $(0, 1)$ and $\left(\sum_{\lambda=1}^{t} \sigma[\lambda, 0], 0\right)$. The algorithm outputs $(tsf, c+1)$ and $\left(tsf + \sum_{\lambda=1}^{t} \sigma[\lambda, 0], c\right)$ as required. If $\sum_{\lambda=1}^{t} \sigma[\lambda, 0] > lim - tsf$, there exists the single point $(0, 1)$. The algorithm outputs $(tsf, c+1)$ as required.

Suppose now that the algorithm is correct for all the costed binary multisequence of length $p^{n'}$ with $n' < n$. Then the computation of the CEJLCS of a costed binary multisequence $\mathcal{S} = (\mathbf{S}, \sigma, p^n)$ calls the two subroutines as follows

(1) CEJLCS $((B(\mathcal{S}), B(\sigma), (p-1)p^{n-1}), tsf, \min\{tsf+T-1, lim\}, c+(p-1)p^{n-1})$, where $T = \sum_{\lambda=1}^{t}\sum_{i=0}^{l-1} \min\{T_{\lambda,i0}, T_{\lambda,i1}\}$. Note that $T > 0$, we have $tsf \le min(tsf+T-1, lim)$.

(2) If $tsf+T \le lim$, we have the CEJLCS $((D(\mathcal{S}), D(\sigma), (p-1)p^{n-1}), tsf+T, lim, c)$, where $T = \sum_{\lambda=1}^{t}\sum_{i=0}^{l-1} \min\{T_{\lambda,i0}, T_{\lambda,i1}\}$. Note that $\min\{tsf+T-1, lim\} = tsf+T-1$ and the inductive hypothesis subroutine (1), the following points list

$$\{(tsf+k_0, c+(p-1)p^{n-1}+LC_{k_0}(B(\mathcal{S}))), \cdots, (tsf+k_v, c+(p-1)p^{n-1}+LC_{k_v}(B(\mathcal{S})))\}$$

are the critical points of $B(\mathcal{S})$, where $(k_i, LC_{k_i}B(\mathcal{S})), k_i \in [0, \min\{T-1, lim-tsf\}], 0 \le i \le v$. By Lemma 4, if $lim - tsf \le T-1$, then these are the all points of the required output. Similarly, by Lemma 4, if $lim - tsf \ge T$, then this is the first portion of the required output. In terms of subroutine (2), the remained critical points are described as follows

$$\{((tsf+T)+k_0, c+LC_{k_0}(D(\mathcal{S}))), \cdots, ((tsf+T)+k_u, c+LC_{k_u}(D(\mathcal{S})))\},$$

where $(k_i,\ LC_{k_i}(D(\mathcal{S}))), k_i \in [0, lim-(tsf+T)], 0 \le i \le u.$

Thus, algorithm CEJLCS $((\mathbf{S}, \sigma, l), tsf, lim, c))$, where $tsf \le$ lim, outputs the list of points

$$\{(tsf+k_0, c+LC_{k_0}(\mathcal{S})), (tsf+k_1, c+LC_{k_1}(\mathcal{S})), \cdots, (tsf+k_u, c+LC_{k_u}(\mathcal{S}))\}$$

where $(k_i,\ LC_{k_i}(\mathcal{S})),\ k_i \in [0, lim-tsf],\ 0 \le i \le u$. These points are all the critical points over CEJLCS of $\mathcal{S} = (\mathbf{S}, \sigma, l)$. Especially, Algorithm CEJLCS $((\mathbf{S}, \sigma, p^n), 0, N, 0)$, where $N = cost(\mathbf{S} \to 0)$, correctly outputs the critical error joint linear complexity of $\mathbf{S}$. □

Proposition 1 *Let $\mathcal{S} = (\mathbf{S}, \sigma, l)$ be a costed binary multisequence and write $N = cost(\mathbf{S} \to 0)$. Then the algorithm CEJLCS ($(\mathbf{S}, \sigma, p^n), 0, N, 0$) correctly outputs the critical error joint linear complexity spectrum $\mathcal{S}$.*

Proof By Theorem 1, as this input, the algorithm outputs the critical points of $\mathcal{S} = (\mathbf{S}, \sigma, l)$ whose first coordinates lie between 0 and N. This is the complete critical error joint linear complexity spectrum of $\mathcal{S}$ because $cost(\mathbf{S} \to 0) = N$ implies that the critical point with the highest first coordinate is $(N, 0)$. □

Remark 3 In Algorithm 1, the conditions $T > 0$ and $tsf + T \ge lim$ are overlapped. A node can be divided into two nodes when two conditions are simultaneously satisfied. Else, the number of subsequent nodes is only one. Moreover, the conditions $\sum_{\lambda=1}^{t} \sigma[\lambda, 0] > 0$ and $tsf + \sum_{\lambda=1}^{t} \sigma[\lambda, 0] \le lim$ are also overlapped. When two conditions are simultaneously satisfied, two critical points can be output. Else, there exits a single critical point.

Now we consider the computational complexity of Algorithm 1. For simplicity, we assume that the entries in the cost vectors are scaled and quantized to be nonnegative integers rather than real numbers.

Remark 4 Algorithm 1 explores part of a binary tree with each node at depth i having bit operations of $\mathcal{O}[t(p-1)p^{n-i}(\log M + i\log p) + tp^{n-i}\log M)]$, where M denotes a maximal integer in entries of the cost multisequence σ. Hence, taking into account the possible doubling in size of the components of the cost multisequences at each depth in the tree, the total bit operations are

$$\mathcal{O}\left\{\sum_{0\leq i\leq n} 2^i[t(p-1)p^{n-i}(\log M + i\log p) + tp^{n-i}\log M]\right\}.$$

By taking $M = 1$, Algorithm CEJLCS can correctly output the entire critical error joint linear complexity spectrum of a binary multisequence with period p^n in $\mathcal{O}(tp^n \log p)$ bit operations, where p is an odd prime and 2 a primitive root modulo p^2.

If directly using the algorithm in [16] to compute the entire error joint linear complexity spectrum of a p^n-periodic multisequence, $\mathcal{O}(t(p-1)p^n\log p^n)$ bit operations are required.

Example 1 Let $\mathbf{S} = (S_1^{27}, S_2^{27}) = (101101101101101101101101011;\ 101010101101010101101010011)$ be a 27-periodic binary multisequence. Then the result for computing the CEJLCS of $\mathbf{S}$ is a list of points as follows $\{(0,27), (1,26), (2,25), (3,24), (4,8), (6,7), (9,6), (13,2), (21,1), (33,0)\}$. The execution of Algorithm CEJLCS $(\mathbf{S}, 0, 33, 0)$ is in Fig. 2.

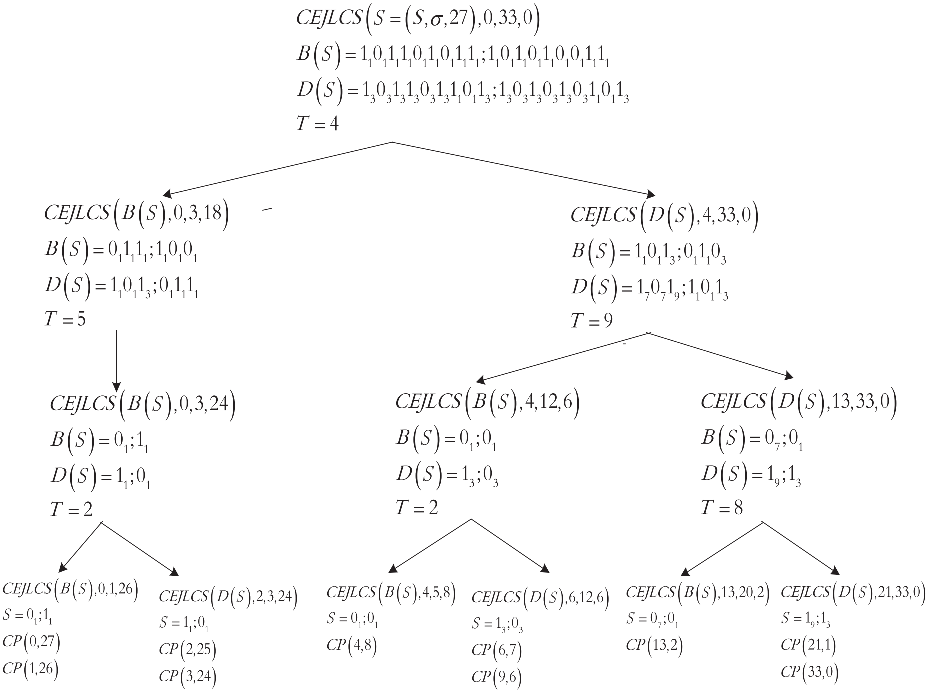

Fig. 2 Schematic execution of Algorithm CEJLCS (**S**,0,33,0)

4 Conclusions

In this paper, we present an efficient algorithm by which the k-error joint linear complexity spectrum for a p^n-periodic binary multisequence can be completely determined using $\mathcal{O}(tp^n \log p)$ bit operations, where p is an odd prime, 2 is a primitive root modulo p^2, and n is a positive integer. Although the algorithm in [16] can also be used to complete the same task, it requires $\mathcal{O}(t(p-1)p^n \log p^n)$ bit operations. Moreover, our results are the further generalization of results in [10]. How to efficiently compute the k-error joint linear complexity spectrum of multisequences with general period remains to be an interesting open problem.

References

1. Berlekamp, E.: Algebraic Coding Theory. McGraw-Hill, New York (1968)
2. Massey, J.: Shift register synthesis and BCH decoding. IEEE Trans. Inf. Theory **15**(1), 122–127 (1969)
3. Stamp, M., Martin, C.F.: An algorithm for the k-error linear complexity of binary sequences with period 2^n. IEEE Trans. Inf. Theory **39**, 1389–1401 (1993)
4. Ding, C., Xiao, G., Shan, W.: The stability theory of stream ciphers. Lecture Notes in Computer Science, vol. 561. Springer, Berlin (1991)
5. Ding, C.: Lower bounds on the weight complexities of cascaded binary sequences, in advances in Cryptology-AVSCRYPT'90. Lecture Notes in Computer Science. In: Seberry, J., Pieprzyk, J. (eds.), vol. 453, pp. 39–43. Springer, Berlin (1991)
6. Niederreiter, H.: Periodic sequences with the large k-error linear complexity. IEEE Trans. Inf. Theory **49**, 501–505 (2003)
7. Games, R., Chan, A.: A fast algorithm for determining the complexity of a binary sequence with period 2^n. IEEE Trans. Inf. Theory **29**(1), 144–146 (1983)
8. Meidl, W.: How many bits have to be changed to decrease the linear complexity ? Des. Codes Cryptogr. **109–122**, 33 (2004)
9. Xiao, G.Z., Wei, S.M., et al.: A fast algorithm for determining the linear complexity of a sequence with period p^n over F_q. IEEE Trans. Inf. Theory **46**(6), 2203–2206 (2000)
10. Lauder, A., Paterson, K.: Computing the error linear complexity spectrum of a binary sequence of period 2^n. IEEE Trans. Inf. Theory **49**(1), 273–280 (2003)
11. Wang, L.P., Zhu, Y.F.: On the lattice basis reduction multisequences synthesis algorithm. IEEE Trans. Inf. Theory **50**(11), 2905–2910 (2004)
12. Sidorenko, V.R., Schmidt G.: A linear algebraic approach to multisequence shift-register synthesis. Probl. Inf. Transm. **47**(2), 149–165 (2011)
13. Xing, C.P., Ding, Y.: Multisequences linear with large field, k-error linear complexity from Hermitian function. IEEE Trans. Inf. Theory **55**(8), 3858–3863 (2009)
14. Niederreiter, H., Venkateswarlu, A.: Periodic multisequences with large error linear complexity. Des Codes Cryptogr. **49**, 33–45 (2008)
15. Meidl, W., Niederreiter, H., Venkateswarlu, A.: Error linear complexity measures for multisequences. J. Complexity **23**(169), 192 (2007)
16. Fulin, L.I., Shixin, Z.H.U.: Computing the k-error joint linear complexity of binary periodic multisequences. The Journal of China Universities of Posts and Telecommunications **20**(6), 96–101 (2013)
17. Sethumadhavana, M., Sindhua, M., Srinivasana, C., Kavithaa, C.: An algorithm for k-error joint linear complexity of binary multisequences. J. Discret. Math. Sci. Cryptogr. **11**(3), 297–304 (2008)