Constructing new covering arrays from LFSR sequences over finite fields

Georgios Tzanakis[a,*], Lucia Moura[b], Daniel Panario[a], Brett Stevens[a]

[a] *School of Mathematics and Statistics, Carleton University*
1125 Colonel By Dr., Ottawa, ON K1S 5B6
[b] *School of Electrical Engineering and Computer Science, University of Ottawa*
800 King Edward Ave., Ottawa, ON K1K 6N5

Abstract

Let q be a prime power and $\mathbb{F}_q$ be the finite field with q elements. A *q-ary m-sequence* is a linear recurrence sequence of elements from $\mathbb{F}_q$ with the maximum possible period. A *covering array $CA(N;t,k,v)$ of strength t* is a $N \times k$ array with entries from an alphabet of size v, with the property that any $N \times m$ subarray has at least one row equal to every possible m-tuple of the alphabet. The *covering array number $CAN(t,k,v)$* is the minimum number N such that a $CA(N;t,k,v)$ exists. Finding upper bounds for covering array numbers is one of the most important questions in this research area. Raaphorst, Moura and Stevens give a construction for covering arrays of strength 3 using m-sequences that improves upon some previous best bounds for covering array numbers. In this paper we introduce a method that generalizes this construction to strengths greater than or equal to 4. Our implementation of this method returned new covering arrays and improved upon 38 previously best known covering array numbers. The new covering arrays are given here by listing the essential elements of their construction.

Keywords: covering arrays, linear feedback shift register sequences, primitive polynomials over finite fields, exhaustive search algorithms
2010 MSC: 94A55, 05B20

1. Introduction

Let M be a $N \times k$ array with entries from an alphabet of size v. If the $N \times s$ subarray of M defined by s columns contains every one of the v^s possible s-tuples at least once, then the set of these columns is *covered*; otherwise, it is *uncovered.* If there exists a positive integer $t \leq k$ such that every t columns of M are covered, then M is a *covering array of strength t and size N*, denoted by $CA(N;t,k,v)$.

[*] Corresponding author

Email addresses: gtzanaki@math.carleton.ca (Georgios Tzanakis), lucia@eecs.uottawa.ca (Lucia Moura), daniel@math.carleton.ca (Daniel Panario), brett@math.carleton.ca (Brett Stevens)

In areas such as software development and manufacturing, it is often infeasible to perform exhaustive system tests. However, empirical research shows that in many types of systems errors are triggered only when a small number of factors interact [18]. In those cases, a practical alternative is *t-way combinatorial testing*, where the objective is to check every t-combination of factors. This approach can dramatically reduce the number of tests that need to be performed, while still being extremely effective in detecting errors [5, 18]. A t-way combinatorial test suite with N tests, for a system with k factors each with v possible levels, corresponds to a $CA(N; t, k, v)$. In this context, the construction of covering arrays of small sizes is important, since it implies a reduction on the number of tests, time and cost needed for a system to be tested.

For given t, k, v, the smallest N such that a $CA(N; t, k, v)$ exists is the *covering array number* $CAN(t, k, v)$. Only few families of covering arrays are known that have a minimum number of rows [3, 16, 17]. Generally, for fixed t and v, a $CA(N; t, k, v)$ with $N = \mathcal{O}(\log k)$ can be constructed in polynomial time [2]. Other upper bounds for covering array numbers follow from numerous methods for obtaining covering arrays that exist in the literature. These include combinatorial and algebraic constructions [4, 21, 22], greedy [2, 7, 31] and metaheuristic [6, 14, 25, 30] computer algorithms, and recursive methods for obtaining new covering arrays from existing ones [7, 10, 11, 15]. Surveys on the subject include [9, 19], and [1, Chapter 3]. Colbourn as of this date actively maintains an online database of the best known upper bounds for covering array numbers [8].

Linear Feedback Shift Register (LFSR) sequences over finite fields have been extensively used in applications such as cryptography and communications [13, 20], but less so for the construction of combinatorial arrays. An *orthogonal array* $OA_\lambda(t, k, v)$ is a $\lambda v^t \times k$ array over an alphabet of size v, with the property that the $\lambda v^t \times t$ subarray defined by any t columns contains each t-tuple *exactly* λ times. When $\lambda = 1$, we simply write $OA(t, k, v)$; such an array is also an optimal $CA(v^t; t, k, v)$. Munemasa [24] uses LFSR sequences corresponding to primitive trinomials over $\mathbb{F}_2$, to create strength-2 binary orthogonal arrays that are very close to being strength 3. Raaphorst et al [27] employ LSFR sequences to construct a $CA(2q^3 - 1; 3, q^2 + q + 1, q)$ for every prime power q. This is, to the best of our knowledge, the only construction in the literature that uses LFSR sequences to construct covering arrays that are not orthogonal arrays.

In this paper, we introduce a method of using LFSR sequences to build covering arrays based on theoretical results established in [27]. It yields covering arrays over finite fields with q elements, strength t, and size $l(q^t - 1) + 1$, where q is a prime power, and l, t are integers with $l \geq 1$, $t \geq 2$. It generalizes two of the previous LFSR-based constructions; for $t = 2$, our method yields an $OA(2, q + 1, q)$, and for $t = 3$ the covering arrays in [27]. For the implementation of our method, we give algorithms that rely on finite field theory and combinatorial exhaustive generation. In particular, we use backtracking for this generation and reduce the search space by proving finite field properties and by using generation of binary necklaces. Finally, we discuss our implementation which gave 38 new covering arrays that improve upon previously best upper bounds for covering array numbers of strength 4, found in [8].

The structure of this paper is as follows. In Section 2 we give some background on

LFSR sequences, we discuss how they are used to construct covering arrays, and we give an overview of our method. The method relies on two parts, to which we dedicate Sections 3 and 4. In Section 5 we discuss our computer implementation of the method and present our results. In Section 6 we conclude with some remarks on connections of our method with finite geometry.

2. LFSR sequences and arrays

2.1. Preliminaries

We begin with some background on LFSR sequences; for a comprehensive presentation we refer to [13] and [20, Chapter 5].

Let $f(x) = x^m + c_{m-1}x^{m-1} + \cdots + c_1 x + c_0 \in \mathbb{F}_{q^m}[x]$, and $I = (b_0, \ldots, b_{m-1}) \in \mathbb{F}_{q^m}$. The sequence $S(f, I) = (a_0, a_1, \ldots)$ defined as

$$a_i = \begin{cases} b_i & \text{if } 0 \leq i < m, \\ -c_{m-1}a_{i-1} - c_{m-2}a_{i-2} - \cdots - c_1 a_{i-(m-1)} - c_0 a_{i-m} & \text{if } i \geq m, \end{cases} \tag{1}$$

is an *LFSR sequence over* $\mathbb{F}_q$ with *characteristic polynomial* f and initial values I. For every such sequence there exists a positive integer P such that $a_i = a_{P+i}$; the smallest such P is the *least period* of the sequence, and it divides $q^m - 1$.

Suppose that f is irreducible, $\alpha \in \mathbb{F}_{q^m}$ is one of its roots, and furthermore α generates the multiplicative group $\mathbb{F}_{q^m}^* = \mathbb{F}_{q^m} \setminus \{0\}$ of $\mathbb{F}_{q^m}$. Then α is a *primitive element of* $\mathbb{F}_{q^m}$, and f is a *primitive polynomial.* An LFSR sequence with primitive characteristic polynomial is an *m-sequence*, that is, a sequence with period $q^m - 1$, which is maximum.

The *trace function* in $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ is the mapping

$$\begin{aligned} \mathrm{Tr}_{q^m/q} : \mathbb{F}_{q^m} &\longrightarrow \mathbb{F}_q \\ x &\mapsto x + x^q + x^{q^2} + x^{q^3} + \cdots + x^{q^{m-1}}, \end{aligned}$$

which is linear over $\mathbb{F}_q$, i.e. $\mathrm{Tr}_{q^m/q}(ca + b) = c\mathrm{Tr}_{q^m/q}(a) + \mathrm{Tr}_{q^m/q}(b)$ for all $c \in \mathbb{F}_q$, $a, b \in \mathbb{F}_{q^m}$. The trace is used to represent m-sequences as follows.

Proposition 2.1 ([20, Theorem 8.33])**.** *Let f be a primitive polynomial over $\mathbb{F}_q$ of degree m and $\alpha \in \mathbb{F}_{q^m}$ one of its roots. For any initial values $I = (a_0, \ldots, a_{m-1})$ there exists a unique element $\beta \in \mathbb{F}_{q^m}$ such that $b_i = \mathrm{Tr}(\beta\alpha^i)$ for all $0 \leq i \leq m-1$. Then, the LFSR sequence $S(f, I) = (a_0, a_1, \ldots)$ has the property that $a_i = \mathrm{Tr}(\beta\alpha^i)$, for all $i \geq 0$.*

In Proposition 2.1, the sequence $\left(\mathrm{Tr}_{q^m/q}(\beta\alpha^i)\right)_{i\geq 0}$ is the *trace representation* of $S(f, T)$. If $\beta = \alpha^s$ for some s, then $\left(\mathrm{Tr}_{q^m/q}(\beta\alpha^i)\right)_{i\geq 0}$ is the left cyclic shift of $(\mathrm{Tr}(\alpha^i))_{i\geq 0}$ by s. In this paper, we refer to $\left(\mathrm{Tr}_{q^m/q}(\alpha^i)\right)_{i\geq 0}$ as *the LFSR sequence associated with* α.

Let α be a primitive element of $\mathbb{F}_{q^m}$, $w = (q^m - 1)/(q - 1)$, and $\mathbf{a} = (a_i)_{i\geq 0}$ be the LFSR sequence associated with α. We denote the left cyclic shift by i of $\mathbf{a}$ as $L_i^w(\mathbf{a}) = (a_i, a_{i+1}, \ldots, a_{i+w-1})$, and define

$$M(\alpha) = \begin{bmatrix} L_0^w(\mathbf{a}) \\ L_1^w(\mathbf{a}) \\ \vdots \\ L_{q^m-2}^w(\mathbf{a}) \\ 0, 0, \dots, 0 \end{bmatrix}. \tag{2}$$

We note that for $0 \leq i \leq q^m - 2$, $0 \leq j \leq w - 1$, the (i, j)-th element of $M(\alpha)$ is $\mathrm{Tr}_{q^m/q}(\alpha^i \alpha^j)$.

The following theorem describes the coverage properties of $M(\alpha)$, and is the cornerstone of the results of this paper.

Theorem 2.2 (See [27, Theorem 2]). *Let q be a prime power, α be a primitive element of $\mathbb{F}_{q^m}$, $m \geq 3$, $w = (q^m - 1)/(q - 1)$, and $c_0, c_1, \dots, c_{w-1}$ denote the column vectors of $M(\alpha)$. Then, the following are equivalent.*

1. *A set of columns $\{c_{i_1}, \dots, c_{i_s}\}$ is uncovered in $M(\alpha)$.*

2. *The elements $\alpha^{i_1}, \dots, \alpha^{i_s} \in \mathbb{F}_{q^m}$ are linearly dependent over $\mathbb{F}_q$.*

Furthermore, if $s = m$ the following is also equivalent to (1) and (2).

3. *There is a row r other than the all-zero, such that $r_{i_1} = \cdots = r_{i_m} = 0$.*

Let q be any prime power, α be a primitive element of $\mathbb{F}_{q^3}$, and let $M'(\alpha)$ be the array that consists of the columns of $M(\alpha)$ in reverse order. Raaphorst et al [27] prove that the vertical concatenation of $M(\alpha)$, $M'(\alpha)$, and a row of zeros, is a $CA(2q^3 - 1; 3, q^2 + q + 1, q)$. The same construction for $m > 3$ does not yield a covering array, so it is natural to ask whether a different generalization can be given.

Observing that reversing the columns of the matrix can be considered as a permutation of its columns, the authors of [27] considered the following generalized construction. They generated $M(\alpha)$ for primitive $\alpha \in \mathbb{F}_{q^m}$, $m \geq 4$, and ran search algorithms to find a permutation group of smallest order $s(m, q)$, such that vertically concatenating the $s(m, q)$ permuted copies of $M(\alpha)$ and a row of zeros, yields a $CA(s(m, q)(q^m - 1) + 1; m, (q^m - 1)/(q - 1), q)$. The resulting arrays in those attempts did not improve the known best upper bounds for covering array numbers.

We note that when $\alpha \in \mathbb{F}_{q^m}$ is primitive then so is α^{-1}, and $M'(\alpha) = M(\alpha^{-1})$. With that in mind, in this paper we consider the following generalization. Let $\alpha_1, \dots, \alpha_l$ be primitive elements of $\mathbb{F}_{q^m}$, and $w = (q^m - 1)/(q - 1)$. We define the *LFSR array generated by* $\alpha_1, \dots, \alpha_l$, denoted $M(\alpha_1, \dots, \alpha_l)$, to be the vertical concatenation of $M(\alpha_1), \dots, M(\alpha_l)$ with all but one copy of the all-zero rows removed. We note that $M(\alpha_1, \dots, \alpha_l)$ is a $(l(q^m - 1) + 1) \times w$ array.

Special cases of LFSR arrays have been previously used to produce covering and orthogonal arrays:

- For any $m \geq 2$, any prime power q, and primitive $\alpha \in \mathbb{F}_{q^m}$, we have that $M(\alpha)$ is an $OA_\lambda(2,(q^m-1)/(q-1),q)$ with $\lambda = q^{m-1}$. This follows from the 2-tuple balance property of m-sequences (see [13, Section 5.6]).

- For $m = 3$, any prime power q, and primitive $\alpha \in \mathbb{F}_{q^m}$, from [27] we have that $M(\alpha, \alpha^{-1})$ is a $CA(2(q^3-1)+1; 3, q^2+q+1, q)$.

- We have from [12] that for $\alpha \in \mathbb{F}_2^m$ whose minimal polynomial is a pentanomial that satisfies certain conditions, the array consisting of $2m$ consecutive columns of $M(\alpha)$ is a $OA_\lambda(3, 2m, 2)$, with $\lambda = 2^{m-3}$, which is a $CA(2^m; 3, 2m, 2)$. This is an extension of a result by Munemasa [24].

- We have from [26] that for $\alpha \in \mathbb{F}_3^m$ whose minimal polynomial is a trinomial that satisfies certain conditions, any array consisting of $3m$ consecutive columns of $M(\alpha)$ is a $OA_\lambda(3, 3m, 3)$, with $\lambda = 3^{m-3}$. This is also a $CA(3^m; 3, 3m, 3)$.

2.2. A new method for constructing covering arrays from LFSR sequences

For integers i, j with $i < j$ we denote $[i, j] = \{i, i+1, \ldots, j\}$.

Let M be an $N \times k$ array with columns $c_0, \ldots, c_{k-1}$, and let $\mathbf{e} \subseteq [0, k-1]$. We define $M[\mathbf{e}]$ to be the subarray of M consisting of the columns c_i, $i \in \mathbf{e}$. The following method yields covering subarrays of LFSR arrays.

Method 1 (Covering arrays from LFSR arrays)**.** Let q be a prime power and $m \geq 3$. The following steps yield a covering array of strength m over $\mathbb{F}_q$.

Step 1: Choose primitive elements $\alpha_1, \ldots, \alpha_l \in \mathbb{F}_{q^m}$ for some $l \geq 2$, and construct $M = M(\alpha_1, \ldots \alpha_l)$.

Step 2: Find a subset $\mathbf{e}$ of column indices of M with maximum size, such that $M[\mathbf{e}]$ is a covering array.

The array $M[\mathbf{e}]$ is a $CA(l(q^m-1)+1; m, |\mathbf{e}|, q)$.

In the following sections we discuss algorithms to efficiently implement this method. In Section 3 we determine how to choose the primitive elements in Step 1 so that we avoid redundancies, and in Section 4 we give an algorithm that solves the optimization problem in Step 2.

3. The choice of primitive elements

A question that naturally arises from the description of Method 1 is what should the choice of $\alpha_1, \ldots, \alpha_l$ be in Step 1, so that the resulting covering array in Step 2 has the maximum number of columns. It is well known (see for example [13, Theorem 4.9]) that there exist $\Phi(q^m-1)/m$ shift-distinct m-sequences of order m over $\mathbb{F}_q$, so this is also the number of different LFSR arrays. Hence, a straightforward approach would be to carry out the method for all $\binom{\Phi(q^m-1)/m}{l}$ l-tuples of elements $\alpha_1,\dots,\alpha_l$ that correspond to those distinct m-sequences.

Let $w=(q^m-1)/(q-1)$, and $q=p^n$ for a prime p and positive integer n. In this section we use finite fields to partition the set of all primitive elements of $\mathbb{F}_{q^m}$ into $\Phi(w/mn)$ classes, with the property that two primitive elements are in the same class if and only if they correspond to LFSR arrays with identical coverage of columns. It follows that we only need to choose one representative from each of those classes, and carry out the method for the $\binom{\Phi(w/mn)}{l}$ l-tuples $\alpha_1,\dots,\alpha_l$ of representatives.

The above-mentioned classes use the notion of cyclotomic cosets. For $i\in\mathbb{Z}_w^*$, we define *the cyclotomic coset of p modulo w that contains i* to be the set

$$C^i_{p,w}=\left\{ip^r \pmod{w}\colon r\in\mathbb{Z}_{\geq 0}\right\}.$$

For any $i,j\in\mathbb{Z}_w^*$, we have that $C^i_{p,w}\subseteq\mathbb{Z}_w^*$ and, furthermore, if $C^i_{p,w}\cap C^j_{p,w}\neq\emptyset$ then $C^i_{p,w}=C^j_{p,w}$. We conclude that there exists a set $\Gamma_{p,w}\subseteq\mathbb{Z}_w^*$ such that

$$\mathbb{Z}_w^*=\bigcup_{i\in\Gamma_{p,w}}C^i_{p,w},\tag{3}$$

and for all $i,j\in\Gamma_{p,w}$ with $i\neq j$, we have that $C^i_{p,w}\cap C^j_{p,w}=\emptyset$. The elements of $\Gamma_{p,w}$ are the *cyclotomic coset leaders of p modulo w*.

Definition 3.1. Let $m\geq 2$, and M_1, M_2 be $N\times k$ arrays with elements from an alphabet of the same finite size. Denote c_i,d_i, $i\in[0,k-1]$ to be their columns, respectively. The arrays *M_1 and M_2 have the same m-coverage* if for every $I\subseteq[0,k-1]$ with $|I|=m$, we have that $\{c_i\colon i\in I\}$ is covered if and only if $\{d_i\colon i\in I\}$ is covered.

The main theoretical result of this section is the following.

Theorem 3.2. *Let q be a power of a prime p, and m,w be integers with $m\geq 2$, $w=(q^m-1)/(q-1)$. Let α be a primitive element of $\mathbb{F}_{q^m}$. Then for any $i\in\Gamma_{p,w}$ we have that α^i is also a primitive element of $\mathbb{F}_{q^m}$, and the following hold.*

1. *For any primitive element $\beta\in\mathbb{F}_{q^m}$, there exists $i\in\Gamma_{p,w}$ such that $M(\beta)$ and $M(\alpha^i)$ have the same m-coverage.*

2. *For all $i,j\in\Gamma_{p,w}$ with $i\neq j$, we have that $M(\alpha^i)$ and $M(\alpha^j)$ do not have the same m-coverage.*

The proof of Theorem 3.2 requires several theoretical results that we give later in the section. First we discuss its implications regarding Method 1. In particular, it follows from Theorem 3.2 that it suffices to carry out Method 1 only for l-tuples of primitive elements $\alpha^{i_1},\dots,\alpha^{i_l}$ where $\{i_1,\dots,i_l\}\in\binom{\Gamma_{p,w}}{l}$. From the next proposition we have that $|\Gamma_{p,w}|=\frac{\Phi(w)}{mn}$, hence we have $\binom{\Phi(w)/mn}{l}$ l-tuples of primitive elements to examine.

Proposition 3.3. *Let p be a prime, and n, q, m, w be integers with $n > 0$, $q = p^n$, $m \geq 2$, $w = (q^m - 1)/(q - 1)$. Then we have that $|C^i_{p,w}| = mn$, for all $i \in \mathbb{Z}^*_w$.*

Proof. The size of $C^i_{p,w}$ is equal to the smallest positive integer r such that $ip^r \equiv i \pmod{w}$, which is equivalent to $w|p^r - 1$, since $\gcd(i, w) = 1$. Now, $w|q^m - 1 = p^{mn} - 1$, and thus $p^{mn} \equiv 1 \pmod{w}$. We note that, by its definition, r is the order of p in $\mathbb{Z}^*_w$ and therefore $p^{mn} \equiv 1 \pmod{w}$ implies that $r|mn$, hence $r \leq mn$. Assume by means of contradiction that $r < mn$. Then, since $r|mn$, it must be $r \leq mn/2$. On the other hand, since $w|p^r - 1$, we have that $p^r - 1 \geq w > p^{mn-n} - 1$, and thus $r > mn - n$. Then $mn/2 > mn - n$ which simplifies to $m < 2$, contradicting our assumption that $m \geq 2$. We conclude that $r = mn$. □

The rest of the section is dedicated to the proof of Theorem 3.2. First, we need to prove a few auxiliary lemmas.

Lemma 3.4. *Let q be a prime power, m be an integer with $m \geq 2$, and α, β primitive elements of $\mathbb{F}_{q^m}$. Then, $M(\alpha)$ and $M(\beta)$ have the same m-coverage if and only if there exists $\gamma \in \mathbb{F}_{q^m}$ such that, for all $s \in [0, q^m - 2]$,*

$$\mathrm{Tr}_{q^m/q}(\alpha^s) = 0 \quad \textit{if and only if} \quad \mathrm{Tr}_{q^m/q}(\gamma\beta^s) = 0.$$

Proof. Let $w = (q^m - 1)/(q - 1)$ and $\{a_0, \ldots, a_{w-1}\}$, $\{b_0, \ldots, b_{w-1}\}$ be the column vectors of $M(\alpha)$ and $M(\beta)$ respectively.

"$\Leftarrow$" Let $I \subseteq [0, w - 1]$ such that $|I| = m$ and $\{a_i : i \in I\}$ is covered. Suppose by contradiction that $\{b_i : i \in I\}$ is uncovered. Then, by Theorem 2.2 there exists a row of $M(\beta)$ that has zeros at the columns b_i, $i \in I$. From the definition of $M(\beta)$, the latter means that there exists $\gamma \in \mathbb{F}^*_{q^m}$ such that $\mathrm{Tr}_{q^m/q}(\gamma\beta^i) = 0$ for all $i \in I$. This implies that $\mathrm{Tr}_{q^m/q}(\alpha^i) = 0$ for all $i \in I$, which means that $\{a_i : i \in I\}$ is uncovered, a contradiction.

"$\Rightarrow$" We observe that $\mathrm{ord}(\alpha^w) = q - 1$, hence α^w is a primitive element of $\mathbb{F}^*_q$, and for every $s \in [0, q^m - 2]$ we have that $\alpha^s = c\alpha^u$, where $u \in [0, w - 1]$, $u \equiv s \pmod{w}$, and $c \in \mathbb{F}_q$. Hence, from the linearity of the trace over $\mathbb{F}_q$, it is sufficient to prove this direction for all $s \in [0, w - 1]$.

We know that $\ker(\mathrm{Tr}_{q^m/q})$ is a vector space over $\mathbb{F}_q$ with dimension $m - 1$ (see for example [23, Theorem 2.1.83]). Since α is primitive, it follows from the above that there exist $i_1, \ldots, i_{m-1} \in [0, w - 1]$ such that $\mathcal{B} = \{\alpha^{i_1}, \ldots, \alpha^{i_{m-1}}\}$ is also a basis for $\ker(\mathrm{Tr}_{q^m/q})$. This means that $\alpha^{i_1}, \ldots, \alpha^{i_{m-1}}$ are linearly independent and the first row of $M(\alpha)$ has zeros at the columns a_i, $i \in \{i_1, \ldots, i_{m-1}\}$. Then, from Theorem 2.2 and our assumption that $M(\alpha)$ and $M(\beta)$ have the same m-coverage, $\beta^{i_1}, \ldots, \beta^{i_{m-1}}$ are also linearly independent, and there exists a row of $M(\beta)$ with zeros at the columns b_i, $i \in \{i_1, \ldots, i_{m-1}\}$. The latter means that there exists $\gamma \in \mathbb{F}^*_{q^m}$ such that $\mathrm{Tr}_{q^m/q}(\gamma\beta^i) = 0$ for all $i \in \{i_1, \ldots, i_{m-1}\}$. We conclude that $\mathcal{B}' = \{\gamma\beta^{i_1}, \ldots, \gamma\beta^{i_{m-1}}\}$ is a basis for $\ker(\mathrm{Tr}_{q^m/q})$.

Now, suppose that $\mathrm{Tr}_{q^m/q}(\alpha^s) = 0$ for some $s \in [0, w - 1]$. Then, by Theorem 2.2, we have that the set of columns $\{a_{i_1}, \ldots, a_{i_{m-1}}, a_s\}$ of $M(\alpha)$ is uncovered and thus the set of columns $\{b_{i_1}, \ldots, b_{i_{m-1}}, b_s\}$ of $M(\beta)$ is also uncovered, from our assumption that $M(\alpha)$ and $M(\beta)$ have the same m-coverage. Hence there exists a row of $M(\beta)$ with zeros at the columns $b_{i_1}, \dots, b_{i_{m-1}}, b_s$, and so there exists $\delta \in \mathbb{F}_{q^m}^*$ such that $\delta\beta^i \in \ker(\mathrm{Tr}_{q^m/q})$ for all $i \in \{i_1, \dots, i_{m-1}, s\}$. Since $\mathcal{B}'$ is a basis for $\ker(\mathrm{Tr}_{q^m/q})$, we have that $\delta = c\gamma$ for some $c \in \mathbb{F}_q^*$. Then $\delta\beta^s \in \ker(\mathrm{Tr}_{q^m/q})$ implies that $\gamma\beta^s \in \ker(\mathrm{Tr}_{q^m/q})$ from the linearity of the trace. □

Lemma 3.5. *Let p be a prime, and q, n, m, l integers with $n > 0$, $q = p^n$, $m \geq 2$, and $0 < l < q^m - 1$. Then, we have that $\mathrm{Tr}_{q^m/q}(x^l) = \mathrm{Tr}_{q^m/q}(x)^l$ for all $x \in \mathbb{F}_{q^m}$ if and only if $l = p^r$ for some integer r such that $0 \leq r < mn$.*

Proof. If $l = p^r$, $0 \leq r < mn$, then $\mathrm{Tr}_{q^m/q}(x^l) = \mathrm{Tr}_{q^m/q}(x)^l$ from the properties of the Frobenius automorphism in $\mathbb{F}_q$.

Conversely, suppose that $\mathrm{Tr}_{q^m/q}(x^l) = \mathrm{Tr}_{q^m/q}(x)^l$ for all $x \in \mathbb{F}_{q^m}$. For a polynomial f on x, we denote $[x^n]f(x)$ to be the coefficient of x^n in f. We observe that

$$[x^{1+(l-1)q}]\mathrm{Tr}_{q^m/q}(x^l) = \begin{cases} 1, & \text{if } l = 1 \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

and

$$[x^{1+(l-1)q}]\mathrm{Tr}_{q^m/q}(x)^l = \begin{cases} 1, & \text{if } l = 1 \\ l, & \text{otherwise.} \end{cases} \tag{5}$$

If $l = 1$, then $l = p^r$ with $r = 0$. If $l > 1$, then it follows from Equations (4) and (5) and our assumption that $\mathrm{Tr}_{q^m/q}(x^l) = \mathrm{Tr}_{q^m/q}(x)^l$, that $l \equiv 0 \pmod{p}$. Hence $l = kp^r$ for some positive integers k, r, with $0 < r < mn$, and $p \not| k$. We have that, for all $x \in \mathbb{F}_{q^m}$,

$$\begin{aligned} \mathrm{Tr}_{q^m/q}(x^k)^{p^r} &= \mathrm{Tr}_{q^m/q}(x^{kp^r}) = \mathrm{Tr}_{q^m/q}(x^l) = \mathrm{Tr}_{q^m/q}(x)^l \\ &= \left(\mathrm{Tr}_{q^m/q}(x)^k\right)^{p^r}. \end{aligned} \tag{6}$$

Taking p^r-th roots in Equation (6) yields that $\mathrm{Tr}_{q^m/q}(x^k) = \mathrm{Tr}_{q^m/q}(x)^k$ for all $x \in \mathbb{F}_{q^m}$. By comparing the coefficients of $\mathrm{Tr}_{q^m/q}(x^k)$ and $\mathrm{Tr}_{q^m/q}(x)^k$ in the same way as we did for $\mathrm{Tr}_{q^m/q}(x^l)$ and $\mathrm{Tr}_{q^m/q}(x)^l$, we have that either $k = 1$, or $k \equiv 0 \pmod{p}$. Since we have assumed that $p \not| k$, it must be $k = 1$, and thus $l = p^r$. □

Lemma 3.6. *Let p be a prime, q, n, m be integers such that $n > 0$, $q = p^n$, $m \geq 2$, and α, β primitive elements of $\mathbb{F}_{q^m}$. Then $M(\alpha)$ and $M(\beta)$ have the same coverage if and only if $\beta = \alpha^{p^r}$, for some $r \in [0, mn - 1]$.*

Proof. If $\beta = \alpha^{p^r}$ for some $r \in [0, mn-1]$, then for every $s \in [0, q^m - 2]$ we have $\mathrm{Tr}_{q^m/q}(\beta^s) = \mathrm{Tr}_{q^m/q}(\alpha^{sp^r}) = \mathrm{Tr}_{q^m/q}(\alpha^s)^{p^r}$. Hence, $\mathrm{Tr}_{q^m/q}(\beta^s) = 0$ if and only if $\mathrm{Tr}_{q^m/q}(\alpha^s) = 0$, and thus $M(\alpha)$ and $M(\beta)$ have the same m-coverage, from Lemma 3.4.

For the converse, assume that $M(\alpha)$ and $M(\beta)$ have the same m-coverage. Since α is primitive, there exists $l \in \mathbb{Z}_{q^m-1}^*$ such that $\beta = \alpha^l$. Then, from Lemma 3.4, there exists $\gamma \in \mathbb{F}_{q^m}$ such that, for all $s \in [0, q^m - 2]$, we have $\mathrm{Tr}_{q^m/q}(\alpha^s) = 0$ if and only if $\mathrm{Tr}_{q^m/q}(\gamma\alpha^{ls}) = 0$. Again from the primitivity of α, we have that $\mathbb{F}_{q^m}^* = \{\alpha^s : s \in [0, q^m - 2]\}$, so we conclude from the above that there exists $\gamma \in \mathbb{F}_{q^m}$ such that, for all $x \in \mathbb{F}_{q^m}^*$, we have

$$\mathrm{Tr}_{q^m/q}(x) = 0 \quad \text{if and only if} \quad \mathrm{Tr}_{q^m/q}(\gamma x^l) = 0. \tag{7}$$

Let y be an element of some extension of $\mathbb{F}_{q^m}$ such that $\mathrm{Tr}_{q^m/q}(\gamma y^l) = 0$. Then $\gamma y^l = z \in \ker(\mathrm{Tr}_{q^m/q}) \subseteq \mathbb{F}_{q^m}$, and $y^l = z/\gamma \in \mathbb{F}_{q^m}$. Since $\gcd(l, q^m - 1) = 1$, the l-th root of z/γ exists, and $y = (z/\gamma)^{1/l} \in \mathbb{F}_{q^m}$. We have proved that $\mathrm{Tr}_{q^m/q}(\gamma x^l)$ splits in $\mathbb{F}_{q^m}$. Now,

$$\mathrm{Tr}_{q^m/q}(\gamma x^l) = \prod_{a \in \ker\left(\mathrm{Tr}_{q^m/q}\right)} \left(\gamma x^l - a\right).$$

Because $\mathrm{Tr}_{q^m/q}(\gamma x^l)$ splits in $\mathbb{F}_{q^m}$, so does $\gamma x^l - a$ for all $a \in \ker(\mathrm{Tr}_{q^m/q})$. Furthermore, the only root of $\gamma x^l - a$ is $(a/\gamma)^{1/l}$, and its degree is l; it follows that it must be $\gamma x^l - a = \gamma(x - (a/\gamma)^{1/l})^l$, and so

$$\mathrm{Tr}_{q^m/q}(\gamma x^l) = \prod_{a \in \ker\left(\mathrm{Tr}_{q^m/q}\right)} \gamma(x - (a/\gamma)^{1/l})^l. \tag{8}$$

From Equation (7) we have that

$$\ker(\mathrm{Tr}_{q^m/q}) = \left\{(a/\gamma)^{1/l}\,;\ a \in \ker(\mathrm{Tr}_{q^m/q})\right\},$$

and it is well known that $|\ker(\mathrm{Tr}_{q^m/q})| = q^{m-1}$, hence Equation (8) becomes

$$\begin{aligned}\mathrm{Tr}_{q^m/q}(\gamma x^l) &= \gamma^{q^{m-1}} \prod_{a \in \ker(\mathrm{Tr}_{q^m/q})} (x-a)^l \\ &= \gamma^{q^{m-1}} \left(\prod_{a \in \ker(\mathrm{Tr}_{q^m/q})} (x-a)\right)^l \\ &= \gamma^{q^{m-1}} \mathrm{Tr}_{q^m/q}(x)^l. \end{aligned} \tag{9}$$

By comparing the coefficient of x^l in $\mathrm{Tr}_{q^m/q}(\gamma x^l)$ and $\gamma^{q^{m-1}}\mathrm{Tr}_{q^m/q}(x)^l$, we have that $\gamma = \gamma^{q^{m-1}}$, which means that $\gamma \in \mathbb{F}_{q^{m-1}}$. However $\gamma \in \mathbb{F}_{q^m}$, hence $\gamma \in \mathbb{F}_{q^m} \cap \mathbb{F}_{q^{m-1}} = \mathbb{F}_q$, and from the linearity of the trace over $\mathbb{F}_q$, $\mathrm{Tr}_{q^m/q}(\gamma x^l) = \gamma \mathrm{Tr}_{q^m/q}(x^l)$. Equation (9) then implies that $\mathrm{Tr}_{q^m/q}(x^l) = \mathrm{Tr}_{q^m/q}(x)^l$, and by Lemma 3.5 we have that $l = p^r$, for some integer r such that $0 \le r < mn$. □

Lemma 3.7. *Let p be prime, and q, n, m integers with $n > 0, q = p^n$, and $w = (q^m - 1)/(q - 1)$. For all $i, j \in \mathbb{Z}^*_{q^m-1}$, we have that $M(\alpha^i)$ and $M(\alpha^j)$ have the same m-coverage if and only if $j \pmod{w} \in C^i_{p,w}$.*

Proof. Suppose that $j \pmod{w} \in C^i_{p,w}$. Then there exist integers r, h such that $j = ip^r + hw$, and thus $\alpha^j = c\alpha^{ip^r}$ with $c = \alpha^{wh}$. We have that $c^{q-1} = \alpha^{w(q-1)h} = \alpha^{(q^m-1)h} = 1$, which means that $c \in \mathbb{F}_q$. Then, from the linearity of the trace and the properties of the Frobenius automorphism we have that, for all positive integers s,

$$\mathrm{Tr}_{q^m/q}(\alpha^{js}) = \mathrm{Tr}_{q^m/q}(c^s \alpha^{isp^r}) = c^s \mathrm{Tr}_{q^m/q}(\alpha^{is})^{p^r}.$$

We conclude that $\mathrm{Tr}_{q^m/q}(\alpha^{js}) = 0$ if and only if $\mathrm{Tr}_{q^m/q}(\alpha^{is}) = 0$, which implies from Lemma 3.4 that $M(\alpha^i)$ and $M(\alpha^j)$ have the same m-coverage.

Conversely, assume that $M(\alpha^i)$ and $M(\alpha^j)$ have the same m-coverage. Then, from Lemma 3.6 we have that $\alpha^j = \alpha^{ip^r}$ for some $r \in [0, mn-1]$, and thus $j \equiv ip^r \pmod{q^m - 1}$. Since $w | q^m - 1$, we also have $j \equiv ip^r \pmod{w}$, which means that $j \pmod{w} \in C^i_{p,w}$. □

We now have the background to give the proof of the main theorem.

Proof of Theorem 3.2. We begin with the first part. Let β be a primitive element of $\mathbb{F}_{q^m}$. From the primitivity of α, we have that there exists $l \in \mathbb{Z}^*_{q^m-1}$ such that $\beta = \alpha^l$. Let $u = l \pmod{w}$. Then $u = l + hw$ for some integer h, and thus $\alpha^u = c\alpha^l$, with $c = \alpha^{hw}$. We have that $c^q = c$, which means that $c \in \mathbb{F}_q$. Hence, for any positive integer s, we have that $\mathrm{Tr}_{q^m/q}(\alpha^{us}) = c^s \mathrm{Tr}_{q^m/q}(\alpha^{ls})$ and therefore $\mathrm{Tr}_{q^m/q}(\alpha^{ls}) = 0$ if and only if $\mathrm{Tr}_{q^m/q}(\alpha^{us}) = 0$. It follows from Lemma 3.4 that $M(\alpha^l)$ and $M(\alpha^u)$ have the same coverage. Since $\gcd(l, q^m - 1) = 1$, then also $\gcd(l, w) = 1$, hence $\gcd(u, w) = 1$ as well. This means $u \in \mathbb{Z}^*_w$ and thus, from Equation (3), there exists $i \in \Gamma_{p,w}$ such that $u \in C^i_{p,w}$. From Lemma 3.7, $M(\alpha^u)$ has the same coverage with $M(\alpha^i)$. Since $M(\alpha^u)$ was shown above to also have the same coverage as $M(\beta)$, we conclude that $M(\beta)$ has the same coverage with $M(\alpha^i)$.

We now prove the second part. Suppose by means of contradiction that $i, j \in \Gamma_{p,w}$, $i \neq j$, and $M(\alpha^i)$ has the same m-coverage with $M(\alpha^j)$. Then, from Lemma 3.7 we have that $j \in C^i_{p,w}$. Thus, $C^i_{p,w} \cap C^j_{p,w} \neq \emptyset$ which means that $C^i_{p,w} = C^j_{p,w}$, as discussed just before Equation (3). This contradicts our assumption that $i, j \in \Gamma_{p,w}$, and we conclude that $M(\alpha^i)$ and $M(\alpha^j)$ do not have the same coverage. □

We close this section by showing how Theorem 3.2 can be used with Method 1. For prime power q, $m \geq 3$, and $l \geq 2$, a $CA(l(q^m - 1) + 1; m, k, q)$ can be found as follows:

1. Create a set $\Gamma_{p,w}$ of cyclotomic coset leaders modulo w, as defined in Equation (3). This can be done by calculating the cosets $C^i_{p,w}$ for all $i \in \mathbb{Z}^*_w$, and picking (any) one representative from each distinct coset.

2. Pick *any* primitive polynomial $\mathbb{F}_q[x]$ and let α be *any* of its roots.

3. For every $\{i_1, \ldots, i_l\} \in \binom{\Gamma_{p,w}}{l}$, find a subset $\mathbf{e}$ of column indices of $M = M(\alpha^{i_1}, \ldots, \alpha^{i_l})$ with maximum size, such that $M[\mathbf{e}]$ is a covering array; see Method 1.

4. Let $(\{i^*_1, \ldots, i^*_l\}, e^*)$ be a pair $(\{i_1, \ldots, i_l\}, e)$ found in Step 3, where e is maximum among all such pairs. We have that $M(\alpha^{i^*_1}, \ldots, \alpha^{i^*_l})[e^*]$ is the desired $CA(l(q^m - 1) + 1; m, k, q)$, with $|k| = |e^*|$.

We note that in Step 2 a different root would produce identical arrays in Step 3. Indeed, for a different root β, we have that $\beta = \alpha^{q^r}$, for some $r \in [0, m-1]$ (see [20, Chapter 3]). Hence, for any i we have $\mathrm{Tr}_{q^m/q}(\beta^i) = \mathrm{Tr}_{q^m/q}(\alpha^{iq^r}) = \mathrm{Tr}_{q^m/q}(\alpha^i)^{q^r} = \mathrm{Tr}_{q^m/q}(\alpha^i)$, where the last equality comes from the fact that $\mathrm{Tr}_{q^m/q}(\alpha^i) \in \mathbb{F}_q$. It follows that $M(\alpha) = M(\beta)$.

4. The search for maximum covering subarrays

Step 2 of Method 1 is an optimization problem whose search space consists of sets of positive integers. In this section we give an algorithm for Step 2. In Section 4.1 we show how the search space can be reduced significantly and generated efficiently, and in Section 4.2 we present a backtracking algorithm to implement Step 2 of Method 1.

4.1. The search space

We begin by introducing a concept required in this section.

Definition 4.1. For $S \subseteq [0, n-1]$ and integer i, we define the *shift of S by i modulo n* to be

$$S +_n i = \{s + i \pmod{n} \colon s \in S\}\,.$$

The next proposition follows from the cyclic nature of LFSR arrays. It can be used to reduce the search space in Step 2 of Method 1.

Proposition 4.2. *Let q be a prime power, $m \geq 2$, $w = (q^m - 1)/(q-1)$, and α be a primitive element of $\mathbb{F}_{q^m}$. Denote by $c_0, c_1, \ldots, c_{w-1}$ the column vectors of $M(\alpha)$ and let $S \subseteq [0, w-1]$. Then, for any $i \in [0, w-1]$, we have that $\{c_j \colon j \in S\}$ is covered if and only if $\{c_j \colon j \in S +_w i\}$ is covered.*

Proof. Assume that $\{c_j \colon j \in S\}$ is not covered. From Theorem 2.2 there exists integer r with $0 \leq r < q^m - 1$ such that the row with index r in $M(\alpha)$ has zeros at the columns with indices from S. From the definition of $M(\alpha)$ this means that $\mathrm{Tr}_{q^m/q}(\alpha^r \alpha^s) = 0$ for all $s \in S$.

For every $j \in S +_w i$ we have that $j = s + i + kw$ for some integer k and some $s \in S$. Setting $c = \alpha^{kw}$ and observing that $c \in \mathbb{F}_q$, we have that $\mathrm{Tr}_{q^m/q}(\alpha^{r-i}\alpha^j) = \mathrm{Tr}_{q^m/q}(c\alpha^r\alpha^s) = c\mathrm{Tr}_{q^m/q}(\alpha^r\alpha^s) = 0$. This shows that the row of $M(\alpha)$ with index $r - i \pmod{q^m - 1}$ has zeros at the columns with indices from $S +_w i$, and using Theorem 2.2, we conclude that the set of these columns is not covered. □

It follows from Proposition 4.2 that it is enough to generate subsets of columns of LFSR arrays that are unique up to cyclic shifts. In Section 4.1.1 we establish a criterion for sets to be unique in this sense using binary necklaces, and in Section 4.1.2 we give an algorithm that generates those unique sets efficiently.

4.1.1. A canonicity criterion for sets

We consider two sets $S, T \subseteq [0, n-1]$ to be isomorphic if there exists integer i such that $T = S +_n i$, and we denote $\mathcal{E}_S$ the equivalence class of S under this isomorphism. We use the notion of binary necklaces to define canonical representatives of those equivalence classes.

Definition 4.3. Let A be an ordered set and $\mathbf{a}$ be a string of elements of A. The *necklace of* $\mathbf{a}$, denoted by neck($\mathbf{a}$), is the lexicographically smallest of all cyclic shifts of $\mathbf{a}$.

In this paper we are only interested in binary necklaces, i.e. $A = \mathbb{F}_2$.

Example 4.4. Let $\mathbf{a} = 10101$. The following are all the cyclic shifts of $\mathbf{a}$, listed in lexicographical order:

$$01011 < 01101 < 10101 < 10110 < 11010,$$

therefore, $\mathrm{neck}(\mathbf{a}) = 01011$. Let $\mathbf{b} = 101010$. All the (distinct) shifts of $\mathbf{b}$ are 010101 and 101010, so $\mathrm{neck}(\mathbf{b}) = 010101$.

For $S \subset [0, n-1]$, we define the *characteristic vector* of S to be $\mathrm{char}_n(S) = b_0 b_1 \cdots b_{n-1} \in \mathbb{F}_2^n$ with $b_i = 1$ if and only if $i \in S$. For a binary string $\mathbf{b} = b_0 \cdots b_{n-1}$ we denote $L^i(\mathbf{b}) = b_i \cdots b_{n-1} b_0 \cdots b_{i-1}$ its left cyclic shift by i. Finally, the all-zero binary string of length r is denoted 0^r. We also introduce the following binary representation for sets.

Definition 4.5. Let n be a positive integer, S a nonempty subset of $[0, n-1]$, and consider $b_i \in \mathbb{F}_2$ with $0 \leq i \leq \max(S)$, such that $b_i = 1$ if and only if $i \in S$. We define

$$\begin{aligned} \mathrm{bin}_n(S) &= 0^{n-\max(S)-1} b_0 \cdots b_{\max(S)} \in \mathbb{F}_2^n \\ &= L^{\max(S)+1}(\mathrm{char}_n(S)). \end{aligned}$$

Moreover, for $S = \emptyset$, we define $\mathrm{bin}_n(S) = 0^n$.

In this paper we select canonical representatives of the above-mentioned equivalence classes as given in the next definition.

Definition 4.6. A set $S \subseteq [0, n-1]$ is *canonical* if either $S = \emptyset$, or S contains 0 and $\mathrm{bin}_n(S)$ is a necklace.

We need to show that this notion of canonicity is well defined. Let $\mathbf{b} = 0^s w$, where w is either a binary string that starts with 1 or the empty string. We define $\mathrm{getSet}(\mathbf{b})$ to be the set whose characteristic vector is $w0^s$.

Proposition 4.7. *Let $S \subseteq [0, n-1]$ be a nonempty set. Then there exists a unique $T \in \mathcal{E}_S$ such that $0 \in T$ and $\mathrm{bin}_n(T)$ is a binary necklace.*

Proof. The characteristic vectors of the elements of $\mathcal{E}_s$ are all the cyclic shifts of the characteristic vector of S, hence there exists a necklace among them. Denote $\mathbf{b}$ that necklace. Then $\mathbf{b} = 0^s w$, for some w that starts with 1, and because $\mathbf{b}$ is a necklace also ends with 1. Let $T = \mathrm{getSet}(\mathbf{b}) = \mathrm{getSet}(0^s w)$. Then T is the set with characteristic vector $w0^s$. This fact implies that $0 \in T$ and that $\mathrm{bin}_n(T) = 0^s w = \mathbf{b}$, which is a necklace. Now, let U be the set in $\mathcal{E}_S$ whose characteristic vector is $\mathbf{b} = 0^s w$, and $\min(U)$ be the least element of U. Then $w0^s$ is also the characteristic vector of $U +_n (-\min(U))$, hence $T = U +_n (-\min(U)) \in \mathcal{E}_S$.

We have shown the existence of the set T in question; it remains to show its uniqueness. Let $T' \in \mathcal{E}_S$ such that $0 \in T'$ and $\mathrm{bin}_n(T')$ is a necklace. Since T and T' are in $\mathcal{E}_S$, we have that $\mathrm{bin}_n(T)$ and $\mathrm{bin}_n(T')$ are cyclic shifts of each other and since they are both necklaces, we have $\mathrm{bin}_n(T) = \mathrm{bin}_n(T')$. Hence $T' = \mathrm{getSet}(\mathrm{bin}_n(T')) = \mathrm{getSet}(\mathrm{bin}_n(T)) = T$. □

We conclude this section by showing that bin_n is a bijection between nonzero binary necklaces and canonical sets.

Proposition 4.8. *There exists a one-to-one correspondence between canonical subsets of* $[0, n-1]$ *and binary necklaces of length* n.

Proof. We claim that bin_n is the needed bijection and getSet is its inverse. For $S = \emptyset$ we have $\text{bin}_n(S) = 0^n$ and $\text{getSet}(0^n) = \emptyset = S$.

Let $S \subseteq [0, n-1]$ be canonical and nonempty. Then by the definition of canonicity, $\text{bin}_n(S)$ is a nonzero necklace of length n. Hence bin_n maps canonical sets to nonzero binary necklaces of length n.

Let $\mathbf{b}$ be a nonzero binary necklace of length n. Then $\mathbf{b} = 0^s w$, $0 \leq s < n-1$ for some w starting with 1. Let $T = \text{getSet}(\mathbf{b})$. Then T has characteristic vector $w0^s$, hence $0 \in T$. Furthermore $\text{bin}_n(T) = \mathbf{b}$, a necklace. We conclude that getSet maps binary necklaces to canonical sets, and it is the inverse of bin_n when that is restricted to canonical sets. □

4.1.2. Generating canonical subsets

Algorithm 1 Generating all nonzero binary necklaces of length n [28].

```
procedure BinaryNecklaces($\mathbf{b}$)
    Output $\mathbf{b}$
    $done \leftarrow$ false
    while not $done$ do
        $\mathbf{b} \leftarrow L(\mathbf{b})$
        $\mathbf{b}' \leftarrow \tau(\mathbf{b})$
        if $b'$ is a necklace then
            BinaryNecklaces($\mathbf{b}'$)
        else
            $done \leftarrow$ true
Main;
BinaryNecklaces($0^{n-1}1$)
```

In this section we use the correspondence between canonical sets and binary necklaces to efficiently generate all canonical subsets of $[0, n-1]$.

For a binary string $\mathbf{b} = b_0 \cdots b_{n-1}$ we denote $\tau(\mathbf{b}) = b_0 \cdots b_{n-2}\overline{b_{n-1}}$, where $\overline{b_{n-1}}$ is the binary complement of b_{n-1}. Algorithm 1 above is from Ruskey et al [28]; it generates every nonzero binary necklace exactly once. An important feature of this generation is that once a non-necklace is encountered, the algorithm backtracks. Thus each time a necklace is generated, exactly one non-necklace is examined, therefore yielding a very efficient algorithm, which requires on average only two "necklace checks" for each necklace generated. In order to generate canonical sets, we translate Algorithm 1 to the language of sets. The next lemma is key to this translation.

Lemma 4.9. *Let* $S \subseteq [0, n-1]$ *be a nonempty set. Then, for all integers* j *with* $1 \leq j < n - \max(S)$*, we have that* $\text{bin}_n(S \cup \{\max(S) + j\}) = \tau\left(L^j(\text{bin}_n(S))\right)$.

Proof. Let $\mathbf{b} = \mathrm{bin}_n(S) = 0^{n-\max(S)-1}b_0 \cdots b_{\max(S)}$, where $b_i \in \mathbb{F}_2$ and $b_i = 1$ if and only if $i \in S$. Then for integer j with $1 \leq j < n - \max(S)$, we have $\tau\left(L^j(S)\right) = 0^{n-(\max(S)+j)-1}b_0 \cdots b_{\max(S)}0^{j-1}1 = \mathrm{bin}_n(S \cup \{\max(S) + j\})$. □

Algorithm 2 Generating all nonempty canonical subsets of $[0, n-1]$.

```
procedure CanonicalSubsets($S$,$n$)
    Output $S$
    for $j$ from $\max(S)+1$ to $n-1$ do
        if bin($S \cup \{j\}$) is a necklace then
            CanonicalSubsets($S \cup \{j\}$,$n$)
        else
            break
Main;
CanonicalSubsets($\{0\}$, $n$)
```

Theorem 4.10. *Algorithm 2 returns all the nonempty canonical subsets of* $[0, n-1]$.

Proof. The proof follows by combining the fact that Algorithm 1 generates all the binary necklaces of length n [28], together with Proposition 4.8 and Lemma 4.9. □

4.2. A backtracking algorithm to search for covering subarrays

In this Section we give a backtracking algorithm to compute Step 2 of Method 1. In particular, the procedure CASearch(M) in Algorithm 3 returns a subset $\mathbf{e}$ of column indices of M with maximum size, such that $M[\mathbf{e}]$ is a covering subarray. This is accomplished by the recursive procedure CASearchBT where canonical sets that correspond to covering subarrays of M are generated (according to Algorithm 2) while searching for one of maximum size.

In CASearchBT we use the framework in Algorithm 2 to go through the canonical subsets of columns of M. At each recursive call, we have a set $\mathbf{e} \subseteq [0, w-1]$ such that $M[\mathbf{e}]$ is a covering array, and we generate a set of candidate columns c such that $M[e \cup \{c\}]$ is a covering array. This set is defined as

$$C_M(\mathbf{e}) = \{c\colon\ \max(\mathbf{e}) < c < w \text{ and } M[e \cup \{c\}] \text{ is a covering array}\}\,.$$

We remark that any set $\mathbf{f}$ such that $\mathbf{e} \subseteq \mathbf{f}$ and $M[\mathbf{f}]$ is a covering array, must satisfy $\mathbf{f} \subseteq \mathbf{e} \cup C_M(\mathbf{e})$. So, we recursively continue with $\mathbf{e} \cup \{c\}$, for every possible $c \in C_M(\mathbf{e})$, as long as it can lead to a maximum sized set and it is canonical.

In Algorithm 3, the global variable MAX stores the largest subset $\mathbf{e}$ found so far, with the desired property. We employ the classical idea of bounding in backtracking by examining the size of $C_M(\mathbf{e})$; more specifically in line 7 we backtrack whenever $|\mathbf{e} \cup C_M(\mathbf{e})| \leq |\mathrm{MAX}|$. Moreover, when we consider extensions of $\mathbf{e}$, a similar condition can be used to limit the choice of elements of $C_M(\mathbf{e})$ in line 9.

Algorithm 3 Algorithm to carry out Step 2 of Method 1

```
 1: procedure CASEARCH($M$)
 2:     procedure CASEARCHBT($\mathbf{e}, C_M(\mathbf{e})$)
 3:         global MAX
 4:         if $|\mathrm{e}| > |\mathrm{MAX}|$ then
 5:             ▷ Record $\mathbf{e}$ if it is the best found so far
 6:             MAX $\leftarrow \mathbf{e}$
 7:         if $|\mathrm{e}| + |C_M(\mathbf{e})| > |\mathrm{MAX}|$ then
 8:             ▷ Otherwise $\mathbf{e}$ cannot be extended to a maximum
 9:             for $i$ from 0 to $|C_M(\mathbf{e})| - (|\mathrm{MAX}| - \mathbf{e})$ do
10:                 ▷ Larger $i$ cannot yield maximum
11:                 $c \leftarrow C_M(e)_i$    ▷ the $i$-th element of $C_M(\mathbf{e})$
12:                 if bin($\mathbf{e} \cup \{c\}$) is a necklace then
13:                     $X \leftarrow$ EXTEND$C_M(e,\ c,\ C_M(\mathbf{e})_{>c}, M)$
14:                     ▷ $X \leftarrow C_M(e \cup \{c\})$
15:                     CASEARCHBT $(\mathbf{e} \cup \{c\}, X)$
16:                 else
17:                     break
18:     global MAX $\leftarrow \{0\}$
19:     CASEARCHBT($\{0\}, \{1, \dots, w-1\}$)
20:     return MAX
```

An important feature of the algorithm is the recursive construction of $C_M(\mathbf{e} \cup \{c\})$ based on $C_M(\mathbf{e})$, accomplished by procedure ExtendC_M called in line 13 of Algorithm 3, and given in Algorithm 4. This is based on a series of definitions and Proposition 4.11, that are given next.

We recall that q is a prime power, $m \geq 4$, $M = M(\alpha_1, \dots, \alpha_l)$ where $\alpha_1, \dots, \alpha_l$ are primitive elements of $\mathbb{F}_{q^m}$, and $w = (q^m - 1)/(q - 1)$. We denote by $c_0, \dots, c_{w-1}$ the columns of M. For $\mathbf{x} = \{x_1, \dots, x_{m-2}\}$ with $0 < x_1 < \dots < x_{m-2} < w$ we define

$$U_M(\mathbf{x}) = \left\{j \colon x_{m-2} < j < w,\ \text{and} \left\{c_0, c_{x_1}, \dots, c_{x_{m-2}}, c_j\right\} \text{is not covered}\right\}.$$

Furthermore, for any positive integer j we denote

$$C_M(\mathbf{e})_{>j} = \{i \colon i \in C_M(\mathbf{e}), i > j\}.$$

The following proposition shows how to compute $C_M(\mathbf{e})$ recursively.

Proposition 4.11. *Let $\mathbf{e} \subseteq [0, m-1]$ such that $M[\mathbf{e}]$ is a covering array. Let $c \in C_M(\mathbf{e})$, and*

$$R(c) = \bigcup_{\{x_1, \dots, x_{m-2}\} \in \binom{\mathbf{e}}{m-2}} \{i + c \colon i \in U_M(x_1 - c, \dots, x_{m-2} - c)\},$$

where all computations are modulo w. Then $C_M(\mathbf{e} \cup \{c\}) = C_M(\mathbf{e})_{>c} \setminus R(c)$.

Algorithm 4 Updating $C_M(\mathbf{e})$ recursively as per Proposition 4.11

```
$c \in C_M(\mathbf{e})$
Returns $C_M(e \cup \{c\})$
procedure EXTEND$C_M$($\mathbf{e}$, $c$, $C_M(\mathbf{e})_{>c}$, $M$)
    if $C_M(\mathbf{e})_{>c} = \emptyset$ then
        return $\emptyset$
    if $n < m - 1$ then
        return $C_M(\mathbf{e})_{>c}$
    else
        $C \leftarrow C_M(\mathbf{e})_{>c}$
        for all $x_1, \ldots, x_{m-2} \in \binom{\mathbf{e}}{m-2}$ do
            $C \leftarrow C \setminus (U_M(x_1 - c, \ldots, x_{m-2} - c) +_w c)$
            if $C = \emptyset$ then
                return $\emptyset$
        return $C$
```

Proof. "$\subset$" Let $d \in C_M(\mathbf{e} \cup \{c\})$. Then $d > c$ and $M[\mathbf{e} \cup \{c, d\}]$ is a covering array. Hence, its subarray $M[\mathbf{e} \cup \{d\}]$ is also a covering array, and therefore $d \in C_M(\mathbf{e})_{>c}$. It remains to show that $d \notin R(c)$.

Assume by means of contradiction that $d \in R(c)$. Then there exist $x_1, \ldots, x_{m-2}, \in \mathbf{e}$, and $i \in U_M(x_1 - c, \ldots, x_{m-2} - c)$, such that $d = i + c$. Then $\{0, x_1 - c, \ldots, x_{m-2} - c, i\}$ is uncovered in M, and from Proposition 4.2 we have that

$$\begin{aligned} \{0, x_1 - c, \ldots, x_{m-2} - c, i\} +_w c &= \{c, x_1, \ldots, x_{m-2}, i + c\} \\ &= \{c, x_1, \ldots, x_{m-2}, d\} \end{aligned}$$

is also uncovered in M. This contradicts our assumption that $d \in C_M(\mathbf{e} \cup \{c\})$.

"$\supseteq$" Let $d \in C_M(\mathbf{e})_{>c} \setminus R(c)$, and assume by means of contradiction that $d \notin C_M(e \cup \{c\})$. Then there exist $x_1, \ldots, x_{m-2} \in \mathbf{e}$ such that the set of columns with indices in $I = \{x_1, \ldots, x_{m-2}, c, d\}$ is uncovered. From Proposition 4.2 we have that the set of columns with indices in $I +_w (-c) = \{x_1 - c, \ldots, x_{m-2} - c, 0, d - c\}$ is also uncovered. Thus, $d - c \in U_M(x_1 - c, \ldots, x_m - c)$. This implies that $d \in R(c)$, a contradiction. □

Procedure EXTENDC_M in Algorithm 4 is based on the recursive computation of $C_M(\mathbf{e} \cup \{c\})$ based on $C_M(\mathbf{e})$ and $R(c)$ given in Proposition 4.11.

The discussion in this section gives the arguments for correctness of our main algorithm, stated in the next theorem.

Theorem 4.12. *Let q be a prime power, $m \geq 3$, $\alpha_1, \ldots, \alpha_l$ primitive elements of $\mathbb{F}_{q^m}$, and $M = M(\alpha_1, \ldots, \alpha_l)$. Then Algorithm 3 returns a canonical subset $\mathbf{e}$ of $[0, w - 1]$ such that $M[\mathbf{e}]$ is a covering array with maximum size.*

5. Implementation and new bounds for covering array numbers

In this section, we construct covering arrays using Algorithm 3, and discuss the results of our search for strength 4 covering arrays. We observe that for $m = 3$ the construction in [27] gives the maximum arrays $M[\mathbf{e}]$ where $\mathbf{e} = [0, w-1]$. Hence, the smallest open case is when $m = 4$. At the end of the section we comment on the search of covering arrays of higher strengths.

Our experiments are as follows. For prime powers q, $2 \leq q \leq 23$, and l up to 6, we choose l-tuples $\alpha_1, \ldots, \alpha_l$ of primitive elements of $\mathbb{F}_{q^4}$ (as per Section 3), and construct $M = M(\alpha_1, \ldots, \alpha_l)$. We then run the procedure CASearchBT in Algorithm 3 with input M, which yields a set $\mathbf{e}$ of indices of columns of M such that $M[\mathbf{e}]$ is a $CA(N; 4, k, q)$, with $N = l(q^4 - 1) + 1$ (by the construction of M), and $k = |\mathbf{e}|$.

In Table 1, the columns denoted $CA(N; 4, k, q)$ contain the parameters of the covering array with the largest k obtained from our experiments for the corresponding values of q and l, and different choices of $\alpha_1 \ldots, \alpha_l$. The number of these choices varies. Entries with an asterisk (*) indicate that all the possible l-tuples were tested, and the procedure CASearchBT was complete. For the rest of the entries, up to 30 l-tuples were tested at random, and for each of them CASearchBT was not run until the end. This means that k is the largest found by our partial runs of CASearchBT, but may not be the optimum one.

q	l	$CA(N;4,k,q)$	PrevN	q	l	$CA(N;4,k,q)$	PrevN
2	2	$CA(31; 4, 6, 2)$*	21	9	2	$CA(13121; 4, 18, 9)$	13113
3	2	$CA(161; 4, 10, 3)$*	159	9	3	$CA(\mathbf{19681}; 4, 42, 9)$	**30537**
3	3	$CA(241; 4, 12, 3)$*	189	9	4	$CA(\mathbf{26241}; 4, 50, 9)$	**30537**
3	4	$CA(321; 4, 12, 3)$*	189	9	5	$CA(\mathbf{32801}; 4, 82, 9)$	**33129**
4	2	$CA(\mathbf{511}; 4, 17, 4)$*	**760**	11	2	$CA(29281; 4, 21, 11)$	29271
4	3	$CA(766; 4, 20, 4)$	760	11	3	$CA(\mathbf{43921}; 4, 37, 11)$	**69091**
4	4	$CA(1021; 4, 20, 4)$	760	11	4	$CA(\mathbf{58561}; 4, 77, 11)$	**69091**
5	2	$CA(\mathbf{1249}; 4, 16, 5)$	**1865**	11	5	$CA(\mathbf{73201}; 4, 125, 11)$	**73931**
5	3	$CA(\mathbf{1873}; 4, 25, 5)$	**2845**	13	2	$CA(57121; 4, 24, 13)$	57109
5	4	$CA(2497; 4, 23, 5)$	1865	13	3	$CA(\mathbf{85681}; 4, 45, 13)$	**136045**
7	2	$CA(4801; 4, 15, 7)$	4795	13	4	$CA(\mathbf{114241}; 4, 98, 13)$	**136045**
7	3	$CA(7201; 4, 26, 7)$	7189	13	5	$CA(\mathbf{142801}; 4, 170, 13)$	**146185**
7	4	$CA(9601; 4, 43, 7)$	9583	16	2	$CA(\mathbf{131071}; 4, 28, 16)$	**188401**
7	5	$CA(12001; 4, 47, 7)$	9583	16	3	$CA(\mathbf{196606}; 4, 55, 16)$	**315136**
8	2	$CA(8191; 4, 17, 8)$	8184	16	4	$CA(\mathbf{262141}; 4, 129, 16)$	**315136**
8	3	$CA(12286; 4, 30, 8)$	12272	17	2	$CA(\mathbf{167041}; 4, 29, 17)$	**240721**
8	4	$CA(\mathbf{16381}; 4, 48, 8)$	**18880**	17	3	$CA(\mathbf{250561}; 4, 61, 17)$	**402577**
8	5	$CA(20476; 4, 65, 8)$	19776	17	4	$CA(\mathbf{334081}; 4, 141, 17)$	**402577**
8	6	$CA(24571; 4, 67, 8)$	19776	19	2	$CA(\mathbf{260641}; 4, 30, 19)$	**377227**
				23	2	$CA(\mathbf{781249}; 4, 35, 23)$	**815167**

Table 1: Overview of our results, where $N = l(q^4 - 1) + 1$. The columns denoted PrevN contain the previous smallest upper bounds for $CAN(4, k, v)$; bold indicates improvement.

We recall that the *covering array number* $CAN(t, k, q)$ is the smallest n such that a $CA(n; t, k, q)$ exists. Hence, a $CA(N; t, k, q)$ implies that N is an upper bound for

$CAN(t,k,q)$. In Table 1, column PrevN contains the previously smallest known [8] upper bounds for $CAN(4,k,q)$, for the k and q of the corresponding array. Bounds in bold indicate that they are improved by our results, and the numbers of rows of the corresponding arrays, also indicated in bold, are the new smallest known upper bounds for $CAN(4,k,q)$.

More results follow recursively; from the fusion operation [10] we obtain a $CA(N-2(v-q);t,k,v)$ from a $CA(N;t,k,q)$. Table 2 contains the results of the fusion operation on our arrays, that improve previously smallest bounds. These are $CA(N;t,k,v)$ with $v \in \{q-1, q-2\}$, $N = l(q^4-1)+1-2(v-q)$, and q,k from the corresponding array in Table 1. We note that these include arrays with alphabets that are not prime powers.

In Table 3 we give the essential elements of the construction of the new arrays displayed in Table 1. The first column contains their parameters $CA(N;t,k,q)$. We recall that the arrays are of the form $M(\alpha_1,\ldots,\alpha_l)[\mathbf{e}]$ for primitive elements $\alpha_1,\ldots,\alpha_l \in \mathbb{F}_{q^4}$, and $\alpha_j = \alpha^{i_j}$, $j=1,\ldots,l$, for powers i_j chosen as per Section 3, for a fixed primitive $\alpha \in \mathbb{F}_{q^4}$. These powers are listed in the second column, and α is a root of the polynomial $P_q(x)$ in Table 4, for the corresponding q.

We demonstrate with an example how to construct the covering arrays in Table 3. For example, to obtain a $CA(1249;4,16,5)$, we generate the LFSR sequences associated with α and α^7, where α is a root of $P_5(x)$, given in Table 4. Then the rows of the array are obtained by generating all the 5^4-1 shifts of the two sequences, choosing for each shift only the elements with indices from $\mathbf{e} = \{0,6,9,15,39,45,48,54,78,84,87,93,117,123,126,132\}$. This array can also be expressed using trace representation, as follows

$$\begin{bmatrix} \left(\mathrm{Tr}_{q^m/q}(\alpha^i)\right)_{i\in\mathbf{e}} \\ \left(\mathrm{Tr}_{q^m/q}(\alpha\alpha^i)\right)_{i\in\mathbf{e}} \\ \vdots \\ \left(\mathrm{Tr}_{q^m/q}(\alpha^{5^4-2}\alpha^i)\right)_{i\in\mathbf{e}} \\ \left(\mathrm{Tr}_{q^m/q}(\alpha^{7i})\right)_{i\in\mathbf{e}} \\ \left(\mathrm{Tr}_{q^m/q}(\alpha\alpha^{7i})\right)_{i\in\mathbf{e}} \\ \vdots \\ \left(\mathrm{Tr}_{q^m/q}(\alpha^{5^4-2}\alpha^{7i})\right)_{i\in\mathbf{e}} \\ 0 \cdots 0 \end{bmatrix}.$$

Table 3: Components of the new covering arrays.

$M(\alpha^{i_1},\ldots,\alpha^{i_l})[\mathbf{e}]$	$i_1,\ldots,i_l$	$\mathbf{e}$
$CA(511;4,17,4)$	1, 31	$5i$, $i=0,1,\ldots,16$
$CA(1249;4,16,5)$	1, 7	0, 6, 9, 15, 39, 45, 48, 54, 78, 84, 87, 93, 117, 123, 126, 132
$CA(1873;4,25,5)$	1, 7,17	0, 9, 12, 21, 24, 33, 36, 45, 48, 57, 60, 69, 72, 81, 84, 93, 96, 105, 108, 117, 120, 129, 132, 141, 144
$CA(16381;4,48,8)$	1, 43, 421, 1324	0-14, 16, 18, 20, 22, 24, 26, 28, 31, 33, 34, 37, 41, 48, 52, 124, 125, 128, 176, 226, 230, 240, 251, 275, 279, 285, 321, 365, 432, 433, 440, 444, 452, 510,

Continued on the next page

Table 3 – *Continued from the previous page*

$M(\alpha^{i_1},\ldots,\alpha^{i_l})[\mathbf{e}]$	$i_1,\ldots,i_l$	$\mathbf{e}$
$CA(19681;4,42,9)$	1, 7,13	$10i$, $i=0,1,\ldots,41$
$CA(26241;4,50,9)$	1, 1129, 1273, 1329	0-3, 5, 6, 8, 9, 11, 13, 15, 16, 18, 19, 22-24, 27-29, 32-34, 38, 43, 46, 49, 54, 56, 57, 60, 65, 67, 70, 80, 97, 102, 117, 168, 201, 226, 310, 335, 358, 367, 369, 391, 458, 468, 482
$CA(32801;4,82,9)$	1, 29, 43, 47, 139	0-81
$CA(43921;4,37,11)$	1, 271, 3491	0, 1, 12, 13, 24, 25, 36, 37, 48, 49, 60, 61, 72, 73, 84, 85, 96, 97, 108, 109, 180, 349, 360, 409, 589, 601, 613, 660, 685, 709, 925, 937, 949, 997, 1020, 1189, 1237
$CA(58561;4,77,11)$	1, 271, 3491, 5861	0-3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 18, 20, 21, 23, 24, 26, 27, 29, 30, 32, 33, 35, 36, 38, 39, 41, 42, 44, 45, 47, 48, 50, 51, 53, 54, 56, 57, 59, 60, 63, 66, 73, 76, 79, 80, 83, 86, 92, 95, 98, 101, 104, 107, 110, 192, 236, 352, 412, 423, 447, 507, 528, 546, 623, 650, 662, 694, 697, 700, 859, 921, 925, 1078, 1254
$CA(73201;4,125,11)$	1,119, 181, 245, 397	0-50, 57-62, 69-74, 81-86, 93-107, 111-114, 176, 177, 197, 230-232, 243, 283, 300, 311, 312, 323, 324, 360-362, 418, 419, 443, 455, 469, 539, 566, 603, 673, 674, 675, 798, 824, 945, 1018, 1066, 1174, 1198, 1308, 1339, 1340
$CA(85681;4,45,13)$	1, 313, 357	0, 1, 14, 15, 28, 29, 42, 43, 56, 57, 70, 71, 84, 85, 98, 99, 112, 113, 126, 127, 140, 141, 154, 168, 182, 238, 336, 532, 574, 686, 714, 742, 798, 1051, 1092, 1162, 1387, 1695, 1737, 1792, 1820, 1862, 1946, 1974, 2030
$CA(114241;4,98,13)$	1, 3, 213, 503	0-38, 42-44, 48, 72-74, 79-81, 83, 84, 123-126, 131, 132, 149, 150, 159, 164, 165, 183, 197, 203, 223, 225, 227, 229, 237, 240, 247, 273, 274, 292, 327, 333, 403, 406, 572, 601, 609, 617, 625, 776, 847, 966, 1115, 1288, 1299, 1359, 1386, 1480, 1669, 1750, 1866, 1952, 2098
$CA(142801;4,170,13)$	1, 79, 109, 171, 421	0-86, 150-169, 243, 245, 247, 264-266, 268, 273, 280, 281, 454, 456, 458-462, 464, 466, 468, 502, 611, 614-619, 642, 773, 782, 797, 803, 810, 811, 828, 829, 965, 975, 977, 979, 983-987, 997, 1158, 1160, 1162, 1163, 1165, 1331, 1447, 1504, 1506, 1643, 1788, 1790, 1792, 2009, 2028, 2152
$CA(131071;4,28,16)$	1, 601	0-3, 5, 6, 8, 11, 12, 17, 22, 23, 25, 36, 45, 46, 50, 157, 184, 352, 661, 1316, 2236, 2736, 3028, 3102, 3126, 3443
$CA(196606;4,55,16)$	1, 4636, 11086	0-3, 5, 6, 8, 11, 12, 17, 20, 22, 26, 29, 34, 35, 39, 40, 45, 49, 54, 69, 73, 78, 91, 100, 102, 105, 111, 120, 122, 137, 146, 155, 164, 184, 208, 239, 332, 333, 395, 399, 404, 537, 598, 858, 1746, 1754, 2020, 2279, 2743, 2751, 2810, 2816, 3189
$CA(262141;4,129,16)$	1, 295, 475, 883	0-53, 87-98, 108-110, 123-125, 129-131, 135-137, 170-173, 182, 185-187, 189-194, 199-201, 210, 223, 308, 337-340, 342, 383, 385, 412, 422, 455, 617, 635, 812, 817, 839, 841, 847, 849, 911, 933, 1438, 1499, 1929, 1938, 1994, 2239, 2758, 2782, 3328, 3383, 3675
$CA(167041;4,29,17)$	1, 18929	0-3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 23, 24, 27, 35, 36, 134, 252, 367, 877, 952, 1771, 1871, 2171, 2239, 3184, 4154

Continued on the next page

Table 3 – *Continued from the previous page*

$M(\alpha^{i_1}, \ldots, \alpha^{i_l})[\mathbf{e}]$	$i_1, \ldots, i_l$	$\mathbf{e}$
$CA(250561; 4, 61, 17)$	1, 6481, 18929	0-3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 18, 20, 21, 23, 24, 26, 27, 29, 30, 32, 33, 35, 36, 38, 40, 41, 43, 46, 49, 52, 54, 57, 60, 82, 93, 98, 110, 115, 120, 123, 151, 168, 194, 219, 248, 264, 371, 709, 910, 1220, 1371, 1428, 1778, 2004, 2324, 2446, 2921, 3623
$CA(334081; 4, 141, 17)$	1, 707, 739, 989	0-61, 63, 65, 67, 69, 71, 73, 102-112, 114, 116, 118, 120, 122, 124, 126, 128, 140, 240, 242, 244, 246, 248, 250, 252, 254, 256-265, 281, 283, 285, 423, 426, 484, 494, 496, 696, 726, 804, 1049, 1127, 1131, 1147, 1149, 1224, 1232, 1237, 1241, 1242, 1245, 1375, 1582, 1913, 2142, 2863, 3061, 3098, 3541, 3576, 3629, 3633, 3863, 3933
$CA(260641; 4, 30, 19)$	1, 32689	0-3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 43, 51, 53, 62, 72, 248, 357, 1470, 1779, 2660, 3200, 4355, 5378, 5756
$CA(781249; 4, 35, 23)$	1, 89	0-3, 5, 6, 8, 9, 11, 12, 14, 18, 19, 21, 22, 24, 25, 27, 28, 31, 35, 41, 45, 118, 347, 586, 1397, 2394, 2505, 4479, 5556, 6315, 8126, 9124, 9954

Although the algorithms in this paper can be applied to search for covering arrays of any strength, the running time increases significantly for strengths $t \geq 5$. We were able to run a few cases of strength $t = 5, 6$ for small alphabets q. We had one notable result, namely a $CA(485; 5, 11, 3)$ which improves the upper bound of $CAN(5, 11, 3)$ from 546 to 485. This array can be constructed as $M(\alpha, \alpha^{17})[\mathbf{e}]$, where α is a root of $x^5 + 2x^4 + 1 \in \mathbb{F}_3[x]$, and $\mathbf{e} = \{11i \colon i = 0, \ldots, 10\}$.

6. Conclusions and future work

We comment on a related backtracking construction of covering arrays by Sherwood, Martirosyan and Colbourn [29], and note that it is similar to ours in that they select length-q^t vectors via linear combinations which are then vertically concatenated to form columns of a covering array. We observe that they also relate coverage with linear independence of these vectors, but consider a more restrictive set of vectors, while allowing more variation on the choice of vectors that are vertically concatenated. We restrict our attention to subarrays that share certain algebraic properties, that is, the ones that come from the same set of columns taken from two different LFSR arrays; its cyclic structure is exploited in our backtrack search via necklace generation. This allows us to find arrays for larger parameter values than in [29]. An open question would be how we could relax the algebraic structure considered to broaden the search space while still being able to handle similarly large parameter values.

On another note, some of our results show patterns that suggest connections with finite geometry. In fact it is established in [27] that the columns of the LFSR array $M(\alpha)$ where α is a primitive element of $\mathbb{F}_{q^m}$, are the points of the projective space $PG(m-1, q)$, and the indices of zeros in every row correspond to the hyperplanes in this projective space. In this context, the columns of the $CA(511; 4, 17, 4)$ in Table 3 are the points of an ovoid (a set of points such that no three are colinear) in that projective space. This suggests that it may

v	q	l	$CA(N;4,k,v)$	PrevN	v	q	l	$CA(N;4,k,v)$	PrevN
10	11	3	$CA(\mathbf{43923}\ ;4,37,10)$	**57486**	15	17	3	$CA(\mathbf{250564};4,61,15)$	**278181**
10	11	4	$CA(\mathbf{58563}\ ;4,77,10)$	**66545**	15	16	2	$CA(\mathbf{131073};4,28,15)$	**173727**
12	13	3	$CA(\mathbf{85683}\ ;4,45,12)$	**114186**	15	16	3	$CA(\mathbf{196608};4,55,15)$	**277827**
12	13	4	$CA(\mathbf{114243}\ ;4,98,12)$	**129345**	15	16	4	$CA(\mathbf{262143};4,129,15)$	**315134**
14	16	2	$CA(\mathbf{131075}\ ;4,28,14)$	**147753**	16	17	2	$CA(\mathbf{167043};4,29,16)$	**188401**
14	16	3	$CA(\mathbf{196610}\ ;4,55,14)$	**226647**	16	17	3	$CA(\mathbf{250563};4,61,16)$	**315136**
14	16	4	$CA(\mathbf{262145}\ ;4,129,14)$	**283193**	16	17	4	$CA(\mathbf{334083};4,141,16)$	**315136**
15	17	2	$CA(\mathbf{167045}\ ;4,29,15)$	**173800**	18	19	2	$CA(\mathbf{260643};4,30,18)$	**355669**

Table 2: Results from the fusion operation on our arrays, where $N = l(q^4-1)+1-2(v-q)$.

q	Minimal polynomial in $\mathbb{F}_q[x]$ of $\alpha \in \mathbb{F}_{q^4}$	
4	$P_4(x) = x^4 + (a+1)x^3 + ax^2 + a,$	$\mathbb{F}_2 = \mathbb{F}_2(a), a^2 = a+1$
5	$P_5(x) = x^4 + x^3 + 2x^2 + 2$	
8	$P_8(x) = x^4 + ax^3 + a,$	$\mathbb{F}_8 = \mathbb{F}_2(a), a^3 = a+1$
9	$P_9(x) = x^4 + ax^3 + a,$	$\mathbb{F}_9 = \mathbb{F}_3(a), a^2 = a+1$
11	$P_{11}(x) = x^4 + 4x^3 + 2$	
13	$P_{13}(x) = x^4 + 6x^3 + 2x^2 + 2$	
16	$P_{16}(x) = x^4 + a^2x^3 + ax^2 + a,$	$\mathbb{F}_{16} = \mathbb{F}_2(a), a^4 = a+1$
17	$P_{17}(x) = x^4 + 6x^3 + 3$	
19	$P_{19}(x) = x^4 + x^3 + 2$	
23	$P_{23}(x) = x^4 + 9x^3 + 5$	

Table 4: Minimal polynomials of the primitive elements used in Table 3.

be possible to obtain a direct construction of covering arrays using finite geometry; we are currently working in this direction.

References

[1] H. Avila George. Constructing Covering Arrays using Parallel Computing and Grid Computing. PhD thesis, Universitat Politècnica de València, 2012.

[2] R. C. Bryce and C. J. Colbourn. A density-based greedy algorithm for higher strength covering arrays. *Software Testing, Verification and Reliability*, 19(1):37–53, 2009.

[3] K. A. Bush. Orthogonal arrays of index unity. *The Annals of Mathematical Statistics*, 23(3):426–434, 1952.

[4] M. Chateauneuf and D. L. Kreher. On the state of strength-three covering arrays. *Journal of Combinatorial Designs*, 10(4):217–238, 2002.

[5] D. M. Cohen, S. R. Dalal, J. Parelius, and G. C. Patton. The combinatorial design approach to automatic test generation. *IEEE Software*, 13(5):83–88, 1996.

[6] M. B. Cohen, C. J. Colbourn, and A. C. H. Ling. Augmenting simulated annealing to build interaction test suites. *14th International Symposium on Software Reliability Engineering (ISSRE 2003)*, pages 394–405, IEEE, 2003.

[7] M. B. Cohen, C. J. Colbourn, and A. C. H. Ling. Constructing strength three covering arrays with augmented annealing. *Discrete Mathematics*, 308(13):2709–2722, 2008.

[8] C. Colbourn. Covering array tables. `http://www.public.asu.edu/~ccolbou/src/tabby/catable.html`. (Retrieved on August 12, 2015)

[9] C. J. Colbourn. Combinatorial aspects of covering arrays. *Le Matematiche (Catania)*, 58(121-167):0–10, 2004.

[10] C. J. Colbourn, G. Kéri, P. R. Soriano, and J.-C. Schlage-Puchta. Covering and radius-covering arrays: constructions and classification. *Discrete Applied Mathematics*, 158(11):1158–1180, 2010.

[11] C. J. Colbourn, S. S. Martirosyan, G. L. Mullen, D. Shasha, G. B. Sherwood, and J. L. Yucas. Products of mixed covering arrays of strength two. *Journal of Combinatorial Designs*, 14(2):124–138, 2006.

[12] M. Dewar, L. Moura, D. Panario, B. Stevens, and Q. Wang. Division of trinomials by pentanomials and orthogonal arrays. *Designs, Codes and Cryptography*, 45(1):1–17, 2007.

[13] S. W. Golomb and G. Gong. Signal Design for Good Correlation: for Wireless Communication, Cryptography, and Radar. Cambridge University Press, 2005.

[14] L. Gonzalez-Hernandez, N. Rangel-Valdez, and J. Torres-Jimenez. Construction of mixed covering arrays of variable strength using a tabu search approach. *Combinatorial Optimization and Applications, Lecture notes in Computer Science*, 6508:51–64, 2010.

[15] A. Hartman. Software and hardware testing using combinatorial covering suites. *Graph Theory, Combinatorics and Algorithms*, pages 237–266, Springer, 2005.

[16] G. O. Katona. Two applications (for search theory and truth functions) of Sperner type theorems. *Periodica Mathematica Hungarica*, 3(1):19–26, 1973.

[17] D. J. Kleitman and J. Spencer. Families of k-independent sets. *Discrete Mathematics*, 6(3):255–262, 1973.

[18] D. R. Kuhn, D. R. Wallace, and A. M. Gallo Jr. Software fault interactions and implications for software testing. *IEEE Transactions on Software Engineering*, 30(6):418–421, 2004.

[19] V. V. Kuliamin and A. A. Petukhov. A survey of methods for constructing covering arrays. *Programming and Computer Software*, 37(3):121–146, 2011.

[20] R. Lidl and H. Niederreiter. Finite Fields, Cambridge University Press, Second edition, 1997.

[21] J. R. Lobb, C. J. Colbourn, P. Danziger, B. Stevens, and J. Torres-Jimenez. Cover starters for covering arrays of strength two. *Discrete Mathematics*, 312(5):943–956, 2012.

[22] K. Meagher and B. Stevens. Group construction of covering arrays. *Journal of Combinatorial Designs*, 13(1):70–77, 2005.

[23] G. L. Mullen and D. Panario. Handbook of Finite Fields. CRC Press, 2013.

[24] A. Munemasa. Orthogonal arrays, primitive trinomials, and shift-register sequences. *Finite Fields and Their Applications*, 4(3):252–260, 1998.

[25] K. J. Nurmela. Upper bounds for covering arrays by tabu search. *Discrete Applied Mathematics*, 138(1):143–152, 2004.

[26] D. Panario, O. Sosnovski, B. Stevens, and Q. Wang. Divisibility of polynomials over finite fields and combinatorial applications. *Designs, Codes and Cryptography*, 63(3):425–445, 2012.

[27] S. Raaphorst, L. Moura, and B. Stevens. A construction for strength-3 covering arrays from linear feedback shift register sequences. *Designs, Codes and Cryptography*, 73(3):949–968, 2014.

[28] F. Ruskey, C. Savage, and T. Min Yih Wang. Generating necklaces. *Journal of Algorithms*, 13(3):414–430, 1992.

[29] G. B. Sherwood, S. S. Martirosyan, and C. J. Colbourn. Covering arrays of higher strength from permutation vectors. *Journal of Combinatorial Designs*, 14(3):202–213, 2006.

[30] T. Shiba, T. Tsuchiya, and T. Kikuno. Using artificial life techniques to generate test cases for combinatorial testing. *Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC 2004)*, pages 72–77, IEEE, 2004.

[31] Y.-W. Tung and W. S. Aldiwan. Automating test case generation for the new generation mission software system. *2000 IEEE Aerospace Conference Proceedings*, volume 1, pages 431–437, IEEE, 2000.