Designs, Codes and Cryptography (2025) 93:5445–5471
https://doi.org/10.1007/s10623-025-01726-5

# Constructions of covering sequences and 2D-sequences

**Yeow Meng Chee[1] · Tuvi Etzion[2] · Hoang Ta[3] · Van Khu Vu[1,4]**

Received: 12 February 2025 / Revised: 13 July 2025 / Accepted: 26 August 2025 /
Published online: 25 September 2025


**Abstract**
An $(n, R)$-covering sequence is a cyclic sequence whose consecutive $n$-tuples form a code of length $n$ and covering radius $R$. Using several construction methods improvements of the upper bounds on the length of such sequences for $n \leq 20$ and $1 \leq R \leq 3$, are obtained. The definition is generalized in two directions. An $(n, m, R)$-covering sequence code is a set of cyclic sequences of length $m$ whose consecutive $n$-tuples form a code of length $n$ and covering radius $R$. The definition is also generalized to arrays in which the $m \times n$ sub-matrices form a covering code with covering radius $R$. We prove that asymptotically there are covering sequences that attain the sphere-covering bound up to a constant factor.

**Keywords** Covering codes · Covering sequences · Covering 2D-sequences · Hamming codes · interleaving · folding

**Mathematics Subject Classification** 05B40

## 1 Introduction

An $(n, R)$***-covering code*** $\mathcal{C}$ is a set of words of length $n$ over a given alphabet $\Sigma_q$, of size $q$, such that each word of length $n$ over $\Sigma_q$ is within distance $R$ from at least one codeword in $\mathcal{C}$. In other words, for each $x \in \Sigma_q^n$, there exists $c \in \mathcal{C}$ such that $d(x, c) \leq R$, where $d(y, z)$, $y, z \in \Sigma_q^n$, denotes the Hamming distance between $y$ and $z$. Covering code were always of interest, but the interest increased due to the following three seminal papers [11, 12, 31]. The interest was also increased partially because of the connection of covering codes to data compression. An excellent book that covers all aspects of such codes is [13].

Chung and Cooper [10] generalized the notion of an $(n, R)$-covering code of length $n$ and radius $R$ to a cyclic sequence whose consecutive $n$-tuples form a covering code of length $n$

Communicated by T. B. Idalino.

✉ Tuvi Etzion
etzion@cs.technion.ac.il

1 Singapore University of Technology and Design, Singapore, Singapore

2 Department of Computer Science, Technion — Israel Institute of Technology, 3200003 Haifa, Israel

3 Hanoi University of Science and Technology, Hanoi, Vietnam

4 College of Engineering and Computer Science, Vin University, Gia Lam, Hanoi 100000, Vietnam

and radius $R$. They called such a structure a de Bruijn covering code since $n$-tuples of a cyclic sequence are considered and this sequence forms a cycle in the de Bruijn graph. However, any cyclic sequence can be viewed as a cycle in the de Bruijn graph, and the sequence itself has a very loose connection to the de Bruijn graph, although some of the constructions in the paper use ingredients and concepts from the de Bruijn graph. Therefore, we prefer to call such a sequence a ***covering sequence***. This is a dual definition of what is known as a robust sequence, i.e., a sequence whose consecutive $n$-tuples form an error-correcting code of length $n$ and minimum distance $d$. Such sequences as well as arrays have been considered for example in [1, 7, 9, 32, 38, 41, 54, 55] and they have variety of applications.

All the discussion from Sect. 3 will be restricted to the binary alphabet and hence alphabet size will rarely be mentioned. However, many of the ideas that will be presented can be generalized quite easily to any alphabet size. An $(n, R)$***-covering sequence*** (an $(n, R)$-CS for short) is a cyclic sequence whose consecutive $n$-tuples form an $(n, R)$-covering code. This structure was first defined and analyzed in [10]. To find the shortest $(n, R)$-CS, denoted by $\mathcal{L}(n, R)$, we define two more related structures of cyclic sequence codes. A cyclic covering sequence code contains sequences in which the consecutive $n$-tuples of all the sequences form an $(n, R)$-covering code. Such codes will be discussed in our exposition.

In recent years many one-dimensional coding problems have been considered in the two-dimensional framework due to modern applications. This is quite natural and has become fashionable from both theoretical and practical points of view. Such generalizations were considered for various structures such as error-correcting codes [2, 45], burst-correcting codes [3, 6, 26, 27], constrained codes [46, 51], de Bruijn sequences [19, 21, 40, 43], M-sequences where the two dimensional sequences are pseudo-random arrays [18, 39]. Other structures designed for applications are robust self-location arrays with window property [7] and structured-light patterns [42]. Therefore, it is very tempting to generalize the concept of covering sequences into a two-dimensional framework and this is one of the targets of the current work.

An $(m \times n, R)$***-covering 2D-sequence*** (an $(m \times n, R)$-C2DS for short) is a doubly-periodic $M \times N$ array (cyclic horizontally and vertically like a torus) over an alphabet of size $q$ such that the set of all its $m \times n$ windows form a covering code with radius $R$.

While in the one-dimensional case we are interested in the $(n, R)$-CS of the shortest length, in the two-dimensional case we are interested in the $(m \times n, R)$-C2DS with the smallest area, but the ratio between $M$ and $N$ can be important too.

***Remark 1*** It is tempting to use the term "covering array" for the two-dimensional matrices, but this term is already reserved to another combinatorial object associated with covering, see [14, 15, 47] and references therein.

Our goals in this paper are to present construction methods for covering sequences, covering sequence codes, and covering 2D-sequences. Some of the constructions yield sequences and codes that are almost optimal.

The rest of the paper is organized as follows. In Sect. 2, some basic results and important known results on covering codes and covering sequences are presented. Section 3 is devoted to cyclic covering sequence codes and the constructions of covering sequences from cyclic covering sequence codes. In Sect. 4 a cyclic covering sequence code based on self-dual sequences is presented. Based on this code a relatively short $(2^k, 1)$-CS whose length is within factor of 1.25 from the sphere-covering bound is obtained. An interleaving construction to obtain an $(n, R)$-CS from an $(n_1, R_1)$-CS and an $(n_2, R_2)$-CS, where $n = n_1 + n_2$ and $R = R_1 + R_2$ is presented in Sect. 5. When $n_1 = n_2$ and $R_1 = R_2$ a better interleaving construction is presented. A construction based on primitive polynomials, with a certain structure, is presented in Sect. 6. Section 7 is the only one devoted to covering 2D-sequences and introduces two methods to generate such arrays: one is by folding a covering sequence and the other is by making all possible shifts of a related covering sequence. These methods are used to obtain upper bounds on the size of such arrays and to construct them. Finally, conclusion and further problems for future research are presented in Sect. 8.

## 2 Preliminaries

The area of covering codes is very well established in information theory as a dual for error-correcting codes. Bounds on the sizes of such codes were extensively studied, where the upper bounds are either by constructions or using probabilistic methods to prove the existence of some codes asymptotically. Lower bounds are usually obtained by analytic methods, but they are usually not much better than the sphere-covering bound which is the most basic bound.

A ball of radius $R$ around a word $\boldsymbol{x}$ of length $n$ is the set of words whose distance is at most $R$ from $\boldsymbol{x}$. The size of such ball denoted by $V_q(n, R)$ is

$$V_q(n, R) = \sum_{i=0}^{R} \binom{n}{i} (q-1)^i \ .$$

Since the balls of radius $R$ around the codewords of an $(n, R)$-covering code $\mathcal{C}$ contain the whole space, it follows that a lower bound on the size of $\mathcal{C}$ is

$$|\mathcal{C}| \geq \frac{q^n}{V_q(n, R)}.$$

This bound is the ***sphere-covering bound***. There exists a covering code that approaches this bound up to a factor roughly $eR \log R$ [37]. The proof method for this bound is probabilistic. A similar bound for an $(n, R)$-CS over a prime power alphabet was presented in [10]. This bound was generalized to any alphabet by Vu [53]. The bound states that for fixed $R$ there exists an $(n, R)$-CS, over $\Sigma_q$, whose length is at most $\mathcal{O}\left(\frac{q^n}{V_q(n,R)} \log n\right)$.

This result of Vu [53] was generalized in our conference paper [8] where we proved the following

**Proposition 2** *Let $m, n$ be nonnegative integers. For any $M \geq m$, there exists an $M \times N$ $(m \times n, R)$-C2DS such that $M \cdot N = \mathcal{O}\left(\frac{q^{mn}}{V_q(mn,R)} \cdot (\log m + \log n)\right)$, for fixed $q$ and $R$.*

The proof of Proposition 2 was presented in [8] using carefully the probabilistic method which was also used in [53] and will not be proved here. The same existence proof can be obtained by applying folding technique which will be presented in Sect. 7 on the one-dimensional sequences which are known to exist by the asymptotic bound of [53].

For small $R$, there are some $(n, R)$-covering codes that attain the sphere-covering bound with equality, such as the Hamming codes of length $n = 2^k - 1$ and radius one. Other codes are very close to the upper bound, such as the ones for $R = 2$ which are perfect asymptotically [50, Construction 4.24] or other similar codes [22, 24]. Similarly, such sparse covering codes were also considered for $R = 3$ [22, 24]. The main goal of the research on $(n, R)$-CSs is to get as close as possible to the upper bounds obtained for covering codes.

One of our constructions will use a span $n$ de Bruijn sequence over $\Sigma_q$. This is a cyclic sequence of length $q^n$, where each $n$-tuple over $\Sigma_q$ is contained in exactly one window of consecutive digits in the sequence. Such sequences exist for all $q \geq 2$ and $n \geq 1$.

Finally, one very trivial bound which was mentioned in [10] will be used in one of the best known bounds (see Table 1).

**Theorem 3** *For any $n, R \geq 1$ we have that $\mathcal{L}(n, R) \leq \mathcal{L}(n+1, R)$.*

## 3 Construction from a cyclic covering code

From this section, the discussion will be focused on binary alphabet, but some of the constructions can be adapted to nonbinary alphabet. This is especially true for this section where all the results can be given also for codes over an arbitrary alphabets. Cyclic covering codes are perhaps the most important ingredients in constructions of $(n, R)$-CSs, especially when $n$ and $R$ are relatively small. They were considered in various papers, e.g. [16, 17, 33, 36]. There are two definitions for covering sequence codes.

An $(n, m, R)$***-CS code*** (an $(n, m, R)$-CSC for short) $\mathcal{C}$ is a set of cyclic codewords of length $m$ such that each word of length $n$ is within distance $R$ from at least one $n$-tuple of a codeword of $\mathcal{C}$, i.e., the consecutive $n$-tuples of all the codewords form an $(n, R)$-covering code.

An $(n, R)$***-CS code*** (an $(n, R)$-CSC for short) $\mathcal{C}$ is a set of cyclic words of possibly different lengths such that each word of length $n$ is within distance $R$ from at least one window of length $n$ in one of the sequences, i.e., the consecutive $n$-tuples of all the codewords form an $(n, R)$-covering code.

The only distinction between the two types of codes is that in the first one of an $(n, m, R)$-CSC, all codewords have the same length $m$ and in the second one of an $(n, R)$-CSC codewords can be of different lengths. The $(n, R)$-CS is a cyclic sequence and hence it is a cyclic covering sequence code with one codeword.

The natural criteria to compare different $(n, R)$-CSCs is the total length of all the sequences in each code. Another criteria for comparison between CSCs which is important in our exposition is the length of the $(n, R)$-CS that can be constructed by combining the codewords of each CSC. This process is done as follows.

The sequences of the code are combined by first concatenating the first $n-1$ bits of each sequence after its last bit. This converts a cyclic sequence to an acyclic sequence, where both sequences have the same $n$-tuples. After the sequences become acyclic, we order the sequences in a cyclic list, where two consecutive sequences in the list overlap such that the suffix of the first sequence aligns with the prefix of the next sequence. The target is to have the total overlaps (sum of the lengths of all the overlaps) as large as possible.

For this purpose, when we concatenate the suffix of length $n-1$ after the last bit of the codeword, we should try to do it for every cyclic shift of the codeword and ultimately use the shifts that yield the largest total overlaps. Merging all the sequences together is done following the order of the list, where each overlap is taken naturally only once. Finally, there might be sequences in the code with periodicity. The periodicity should be eliminated either within the code or in the final $(n, R)$-CS. This is demonstrated in the following example.

***Example 4*** The following eight codewords form a $(9, 10, 1)$-CSC $\mathcal{C}$:

$$[1000010000], [0001001101], [1001111001], [1111010111],$$
$$[1010101010], [0101011000], [0110111001], [0111010000].$$

We extend each cyclic sequence by eight bits to obtain all the 9-tuples of the cyclic sequences in acyclic sequences. We furthermore find the order of these eight sequences in a way that the total overlap between suffixes and prefixes of the consecutive sequences (including the last sequence in the list and the first sequence in the list) will be large as possible, where in this process all possible shifts of all the sequences are considered. The obtained eight acyclic sequences of length 18 are as follows, where the overlap is written after the sequence:

| *the sequence* | *overlap* |
|---|---|
| 100001000010000100 | 6 |
| 000100110100010011 | 5 |
| 100111100110011110 | 5 |
| 111101011111110101 | 5 |
| 101010101010101010 | 5 |
| 010101100001010110 | 4 |
| 011011100101101110 | 5 |
| 011101000001110100 | 3 |

The total number of bits in the eight sequences is 144 and the total number of overlaps is 38. This yield a (9, 1)-CS of length $144 - 38 = 106$ as follows:

$$[10000100001000010011010001001111001100111101011111110$$
$$10101010101010101100001010110111001011011101000001110]$$

However, the codeword [1000010000] of $\mathcal{C}$ is periodic and contains only five 9-tuples as the cyclic sequence [10000]. Moreover, the codeword [1010101010] of $\mathcal{C}$ is also periodic and it contains only two 9-tuples as the cyclic sequence [10]. Hence, they are redundant in the code and in the final (9, 1)-CS. Therefore, we can eliminate the redundancy in the covering sequence and obtain the following (9, 1)-CS of length $106 - 13 = 93$:

$$[100001000010011010001001111001100111101011111110$$
$$101010101100001010110111001011011101000001110]$$

Introducing the codewords [10000] instead of [1000010000] and [10] instead of [1010101010] in the code $\mathcal{C}$ yields a (9, 1)-CSC with 6 codewords of length 10, one codeword of length 5, and one codeword of length 2.

This sequence of length 93 can be extended to a (9, 1)-CS of length 102 with 8 consecutive *ones* which will be required later:

$$[100001000010011010001001111001100111101011111111011111110$$
$$101010101100001010110111001011011101000001110]$$

□

Many of the shortest $(n, R)$-CSs for small $n$ and $R$ were found by computer search. For this purpose we use the well-known problem of the ***Shortest Cyclic Superstring (SCS)***. This problem is designed to find the shortest cyclic superstring, i.e. to find a short sequence as possible which contains all the sequences of a given set $\mathbb{S}$ as subsequences. The acyclic version of the SCS problem, which find the shortest superstring, has been extensively studied in the literature. This problem has numerous applications in data compression [49] and computational biology [29]. However, the problem is known to be NP-complete [28] and even

APX-complete [4], indicating that polynomial-time approximation schemes with arbitrary constant approximations are not to be expected.

Many greedy and approximation algorithms have been developed to tackle this problem, and they have demonstrated numerical efficiency [34, 35, 48]. Furthermore, we note that any solution for the acyclic SCS problem can also be considered a feasible solution for the cyclic SCS problem. In our variant to the problem we built a complete directed graph which represent all the sequences of the code $\mathcal{C}$ and their possible overlaps with all the other sequences that follow them. For this purpose we use the well-known approximation algorithm for the set cover problem [52, Chapter 2].

While comparing codes with codewords of different lengths is not trivial, it is much easier to compare codes in which all codewords have the same length. Given two $(n, m, R)$-CSCs, it is obvious that the code with a smaller number of codewords will be considered better. Given two codes, $\mathcal{C}_1$ an $(n, m_1, R)$-CSC and $\mathcal{C}_2$ an $(n, m_2, R)$-CSC where the total length of the codewords is smaller in $\mathcal{C}_1$ and $m_1 \geq m_2$, then $\mathcal{C}_1$ is considered superior to $\mathcal{C}_2$. However, if the total length of the codewords in $\mathcal{C}_1$ is smaller and $m_1 < m_2$, then the two codes are incomparable. Nevertheless, there should be some tradeoff between the two parameters. We demonstrate these concepts for a length $n = 2^k - 1$ in this section and for a length $n = 2^k$ in Sect. 4.

For a length $n = 2^k - 1$ we consider the Hamming code $\mathcal{C}$. The Hamming code $\mathcal{C}$ of length $2^k - 1$ can be represented as a cyclic covering code that contains $2^{2^k-k-1}$ codewords. If $c$ is a codeword in $\mathcal{C}$ then all the cyclic shifts of $c$ are also codewords in $\mathcal{C}$. Each codeword of $\mathcal{C}$ has $d$ distinct cyclic shifts, where $d$ is a divisor of $n$. For example, the all-zero word is a codeword and it has one distinct cyclic shift. From the set of $d$ distinct cyclic shifts only one is considered in the code.

***Example 5*** For $n = 15$, we consider the cyclic Hamming code of length 15. It has 134 codewords of length 15, 6 codewords of length 5, 2 codewords of length 3, and 2 codewords of length 1. Together, these 144 codewords contain 2048 words of length 15 which form the Hamming code of length 15. They are merged to form a (15, 1)-CS of length 3516 (see Appendix C) compared to a lower bound 2048 obtained by the sphere-covering bound. □

In general, the Hamming code $\mathcal{C}$ contains $2^{2^k-k-1}$ codewords. There are at most

$$\sum_{d|n,\ d<n} 2^d < 2^{\lfloor \frac{n}{2} \rfloor + 1} = 2^{2^{k-1}}$$

cyclic words that contain less than $n$ distinct cyclic shifts (see [20, p. 105, Lemma 3.20]). Thus, the $(2^k - 1, 2^k - 1, 1)$-CSC code contains much less than $\frac{2^{2^k-k-1}}{2^k-1} + 2^{2^{k-1}}$ codewords and hence asymptotically it is an optimal CSC.

We can make a more precise computation when $n = 2^k - 1$ is a prime. In this case only two codewords, the all-zero codeword and the all-one codeword, have less than $n$ distinct shifts. Therefore, the number of codewords in the $(2^k - 1, 2^k - 1, 1)$-CSC is $\frac{2^{2^k-k-1}-2}{2^k-1} + 2$. To obtain a $(2^k - 1, 1)$-CS we have to add to each sequence at most $2^k - 2$ bits to convert it to acyclic sequence which contains all its words of length $2^k - 1$. The $(2^k - 1, 1)$-CS that is generated has length shorter than $2^{2^k-k}$, i.e., within factor of at most 2 from optimality.

## 4 Construction from self-dual sequences

In this section, we continue with the construction of cyclic codes as presented in Sect. 3 by using a specific family of sequences called self-dual sequences. A self-dual sequence is a cyclic sequence $\mathcal{S}$ whose complement $\bar{\mathcal{S}}$ is the same sequence as $\mathcal{S}$ after another shift. The construction will be described for $n$ that it is a power of two, but it can be described to other values of $n$. For a given word of length $n$ we are interested in self-dual sequences of length $2n$. Such sequences are of the form $[X\ \bar{X}]$, where $X$ is a sequence of length $n$ and $[X\ \bar{X}]$ has no periodicity since its length is a power of 2.

The construction for self-dual sequences starts with a $(n, 2n, 1)$-CSC $\mathcal{C}$, where $n = 2^k$, (with $2^{2^{k+1}-k-2}$ $n$-tuples), in which for each $n$-tuple $U$ of a codeword in $\mathcal{C}$ there exists exactly one other $n$-tuple $V$ of a codeword in $\mathcal{C}$ such that $d(U, V) = 1$. For this purpose, the code $\mathcal{C}$ will be designed in such a way that if $[X\bar{X}]$ is a codeword in $\mathcal{C}$, then there exists exactly one codeword $[Y, \bar{Y}]$ in $\mathcal{C}$ such that $d(X, Y) = 1$. This immediately implies the following observation proved also in [5].

**Lemma 6** *The $n$-tuples in the codeword of $\mathcal{C}$ form an $(n, 1)$-covering code $\mathcal{C}$ for which each codeword $c \in \mathcal{C}$ there exists exactly one other codeword $c' \in \mathcal{C}$ such that $d(c, c') = 1$.*

Now, we are in a position to present a construction for an $(n, 1)$-CS based on self-dual sequences of length $2n$.

**Construction 1** Let $\mathcal{C}_n$ be an $(n, 2n, 1)$-CSC code, $n = 2^k$, with $2^{2^k-2k-1}$ sequences, all of them are self-dual. Moreover, if $c \in \mathcal{C}$ then there exists a shift $[X\ \bar{X}]$ of $c$ for which $\mathcal{C}$ has exactly one other codeword $[X'\ \bar{X}']$, where $X'$ differs from $X$ only in the last coordinate.

Let $\mathcal{E}_n$ be the set of all $2^{n-2}$ even-weight words in $\mathbb{F}_2^n$ that start with a *zero*. Let $\mathcal{C}_{2n}$ be the code with self-dual sequences of length $4n$ defined by

$$\mathcal{C}_{2n} \triangleq \{[U\ U + X\ \bar{U}\ \bar{U} + X]\ :\ U \in \mathcal{E}_n,\ [X\ \bar{X}] \in \mathcal{C}_n]]\},$$

where the basis is

$$\mathcal{C}_8 = [0001101111100100], [0001101011100101]$$

□

The computation of the sizes of the codes and the correctness of the details can be found in [5], where the following result was proved.

**Theorem 7** *The code $\mathcal{C}_{2n}$, $n = 2^k$, of Construction 1 is an $(2n, 4n, 1)$-CSC with $2^{2^{k+1}-2k-3}$ codewords which are self-dual sequences of length $2^{k+1}$. If a codeword is in $\mathcal{C}_{2n}$ then it has the form $[X\ \bar{X}]$ and $[X'\ \bar{X}']$ is another codeword in $\mathcal{C}_{2n}$.*

***Remark 8*** Construction 1 is very similar to the constructions presented in [23, 25]. The construction based on self-dual sequences was found by one of the authors and was motivated by the introduction of nearly-perfect covering codes [5]. It was first introduced for the earlier conference version of this paper [8].

From the two codewords of $\mathcal{C}_8$, we can form the optimal $(8, 1)$-CS of length 32

$$[0001101111100100\ 0001101011100101]\ .$$

Construction 1 is applied on $\mathcal{C}_8$ to obtain a (16, 32, 1)-CSC with 128 codewords of length 32. These 128 codewords are partitioned into 64 pairs of the form

$$[X\ \bar{X}] \quad \text{and} \quad [X'\ \bar{X}'] \,.$$

Each pair is combined into one sequence of length 64 that contains all the 16-tuples of the original two codewords of length 32. Hence, this code is an optimal (16, 64, 1)-CSC of length 64 with 64 codewords. Trivially, each sequence can be extended with its first 15 bits to an acyclic sequence of length $64 + 15 = 79$. In other words, these 64 sequences can be concatenated to form a (16, 1)-CS of length $64 \cdot (64 + 15) = 5056$. However, we can merge these sequences with overlaps between the suffixes and the prefixes of the sequences and obtain a (16, 1)-CS of length 4462 (see Appendix D), where the known lower bound, for a (16, 1)-covering code, is 4096. The same idea can be applied recursively on the 64 pairs of sequences of length 32 to obtain an upper bound on the shortest length of a $(2^k, 1)$-CS.

For $n = 2^k$, $k > 2$, we obtain a $(2^k, 2^{k+2}, 1)$-CSC of length $2^{k+2}$ with $2^{2^k-2k-2}$ codewords. This code is optimal as the total length of the codewords is the same as the number of codewords in an optimal $(2^k, 1)$-covering code.

Without considering any overlap we can merge these $2^{2^k-2k-2}$ sequences, whose length is $2^{k+2} + 2^k - 1$ (the $n - 1$ prefix is concatenated after the last bit of each sequence) as an acyclic sequence, to obtain a $(2^k, 1)$-CS of length $2^{2^k-2k-2}(2^{k+2} + 2^k - 1)$. We can also compute some overlaps between the prefixes and the suffixes of these sequences; however, we omit this computation due to its complexity. The number of codewords in an optimal $(2^k, 1)$-covering code is $K = 2^{2^k-k}$, so this sequence has a length of less than $1.25K$, i.e., within factor of at most 1.25 from optimality.

## 5 Interleaving of covering sequences

Interleaving is probably the most simple method to construct $(n, R)$-CSs with a large length and a large covering radius from two or more covering sequences of smaller length and a smaller covering radius. When two covering sequences participate in the construction, one is an $(n_1, R_1)$-CS of length $k_1$ and the other is an $(n_2, R_2)$-covering sequence of length $k_2$, where $n_2 \le n_1 \le n_2 + 1$ and $\gcd(k_1, k_2) = 1$. From these two covering sequences we generate an $(n_1 + n_2, R_1 + R_2)$-CS of length $2k_1k_2$.

**Construction 2** Let $\mathcal{A} = [a_0, a_1, \ldots, a_{k_1-1}]$ be an $(n_1, R_1)$-CS and $\mathcal{B} = [b_0, b_1, \ldots, b_{k_2-1}]$ be an $(n_2, R_2)$-CS and assume further that $n_1 = n_2$ or $n_1 = n_2 + 1$ and $\gcd(k_1, k_2) = 1$. The interleaving of $\mathcal{A}$ and $\mathcal{B}$, is the sequence $\mathcal{S} = [s_0, s_1, \ldots, s_{2k_1k_2-1}]$ of length $2k_1k_2$ defined by $s_0 = a_0$, $s_1 = b_0$, $s_2 = a_1$, $s_3 = b_1$, and in general $s_{2i} = a_i$, where $i$ is taken modulo $k_1$ and $s_{2i+1} = b_i$, where $i$ is taken modulo $k_2$, for $0 \le i \le k_1k_2 - 1$. □

**Theorem 9** *If $n_1 = n_2$ or $n_1 = n_2 + 1$, then the sequence $\mathcal{S}$ defined in Construction 2 is an $(n_1 + n_2, R_1 + R_2)$-CS of length $2k_1k_2$.*

***Proof*** First note that since $\gcd(k_1, k_2) = 1$, it follows that for each $i$ and $j$, $0 \le i \le k_1 - 1$, $0 \le j \le k_2 - 1$ the string $a_i b_j a_{i+1} b_{j+1} \cdots a_{i+n_1} b_{j+n_1} a_{i+n_1+1}$, where indices of $\mathcal{A}$ are taken modulo $k_1$ and indices of $\mathcal{B}$ are taken modulo $k_2$, is a subsequence in $\mathcal{S}$. This also implies that the length of the sequence is $2k_1k_2$. We distinguish between the two cases of $n_1 = n_2$ and $n_1 = n_2 + 1$.

**Case 1** If $n_1 = n_2 = n$, then consider a word $(\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_n, \beta_n)$. Since $\mathcal{A}$ is an $(n, R_1)$-CS and $\mathcal{B}$ is an $(n, R_2)$-CS, it follows that there exists a subsequence $a_i a_{i+1} \cdots a_{i+n-1}$

of $\mathcal{A}$ whose distance from $\alpha_1, \alpha_2, \ldots, \alpha_n$ is at most $R_1$ and there exists a subsequence $b_j b_{j+1} \cdots b_{j+n-1}$ of $\mathcal{B}$ whose distance from $\beta_1, \beta_2, \ldots, \beta_n$ is at most $R_2$. Therefore, the subsequence of $\mathcal{S}$, $a_i b_j a_{i+1} b_{j+1} \cdots a_{i+n-1} b_{j+n-1}$ is within distance at most $R_1 + R_2$ from $(\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_n, \beta_n)$.

**Case 2** If $n_1 = n_2+1$, then consider a word $(\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_{n_2-1}, \beta_{n_2-1}, \alpha_{n_2}, \beta_{n_2}, \alpha_{n_1})$. Since we have that $\mathcal{A}$ is an $(n_1, R_1)$-CS and $\mathcal{B}$ is an $(n_2, R_2)$-CS, it follows that there exists a subsequence $a_i a_{i+1} \cdots a_{i+n_1-1}$ of $\mathcal{A}$ whose distance from $\alpha_1, \alpha_2, \ldots, \alpha_{n_1}$ is at most $R_1$ and there exists a subsequence $b_j b_{j+1} \cdots b_{j+n_2-1}$ of $\mathcal{B}$ whose distance from $\beta_1, \beta_2, \ldots, \beta_{n_2}$ is at most $R_2$. Therefore, the subsequence of $\mathcal{S}$, $a_i b_j a_{i+1} b_{j+1} \cdots a_{i+n_2-1} b_{j+n_2-1} a_{i+n_1-1}$ is within distance at most $R_1 + R_2$ from the word $(\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_{n_2}, \beta_{n_2}, \alpha_{n_1})$. □

***Example 10*** The following $(n, R)$-CSs were obtained by using Construction 2 on two shorter covering sequences which are either trivial or presented in Appendix A or in Appendix B.

Consider the $(9, 0)$-CS of length 512 and a $(9, 1)$-CS length 93. Interleaving these two sequences using Construction 2 yields a $(18, 1)$-CS of length $2 \cdot 512 \cdot 93 = 95232$.

Consider the $(10, 0)$-CS of length 1024 and a $(10, 1)$-CS length 175. Interleaving these two sequences using Construction 2 yields a $(20, 1)$-CS of length $2 \cdot 1024 \cdot 175 = 358400$.

Consider an $(8, 1)$-CS of length 32 and a $(9, 1)$-CS length 93. Interleaving these two sequences using Construction 2 yields a $(17, 2)$-CS of length $2 \cdot 32 \cdot 93 = 5952$.

Consider an $(8, 1)$-CS of length 37 and an $(8, 2)$-CS length 14. Interleaving these two sequences using Construction 2 yields a $(16, 3)$-CS of length $2 \cdot 37 \cdot 14 = 1036$.

Consider an $(8, 1)$-CS of length 37 and a $(9, 2)$-CS length 20. Interleaving these two sequences using Construction 2 yields a $(17, 3)$-CS of length $2 \cdot 37 \cdot 20 = 1480$.

Consider a $(9, 1)$-CS of length 93 and a $(9, 2)$-CS length 20. Interleaving these two sequences using Construction 2 yields a $(18, 3)$-CS of length $2 \cdot 93 \cdot 20 = 3720$.

Consider a $(9, 1)$-CS of length 93 and a $(10, 2)$-CS length 38. Interleaving these two sequences using Construction 2 yields a $(19, 3)$-CS of length $2 \cdot 93 \cdot 38 = 7068$.

Consider a $(10, 1)$-CS of length 175 and a $(10, 2)$-CS length 38. Interleaving these two sequences using Construction 2 yields a $(20, 3)$-CS of length $2 \cdot 175 \cdot 38 = 13300$. □

Construction 2 can be generalized to three or more sequences. When $t$ sequences are interleaved the requirement is that each sequence is an $(n_i, R_i)$-CS of length $k_i$, $1 \leq i \leq t$, where $\gcd(k_i, k_j) = 1$, $1 \leq i < j \leq t$, and $n \leq n_i \leq n+1$ for some positive integer $n$. From these $t$ sequences we generate an $(\sum_{i=1}^{t} n_i, \sum_{i=1}^{t} R_i)$-CS sequence of length $\ell = t \prod_{i=1}^{t} k_i$, $[s_0 s_1 \cdots s_{\ell-1}]$, where $s_r$, $r \equiv j \pmod{t}$, $1 \leq j \leq t$, is a bit from the $j$-th sequence. Unfortunately, the factor $t$ in the length of the sequence makes this construction quite weak for $t > 2$ and also for $t = 2$ (see Construction 2) it is not too effective, but some of our best $(n, R)$-CSs for $n \leq 20$ and $1 \leq R \leq 3$ are obtained by this construction. Fortunately, we can get rid of the factor 2 when $t = 2$ and $n_1 = n_2$ at the expense of some extra small redundancy as follows.

**Construction 3** Let $\mathcal{A} = [a_0, a_1, \ldots, a_{k-1}]$ be an $(n, R)$-CS in which $a_i = 0$ for $0 \leq i \leq n-2$, i.e., $\mathcal{A}$ has a subsequence with $n-1$ consecutive *zeros* (the same can be applied for the *ones*). If $k$ is even then form the following sequence $\mathcal{S}$ presented in $k/2$ parts. The first part is

$$a_0 a_0 a_1 a_1 \; \cdots \; a_{k-1} a_{k-1} a_0 0 \; .$$

The second part is

$$a_1 a_0 a_2 a_1 \; \cdots \; a_{k-1} a_{k-2} a_0 a_{k-1} a_1 0 \; .$$

The $i$-th part, $1 \leq i \leq k/2$, is

$$a_{i-1}a_0a_ia_1a_{i+1}a_2 \ \cdots \ a_{i-2}a_{k-1}a_{i-1}0 \ .$$

These $k/2$ parts, each one of length $2k+2$, are concatenated together, in their order, to the final sequence $\mathcal{S}$.

If $k$ is odd, then form the same parts, where the last one is for $i = \frac{k+1}{2}$. □

**Theorem 11** *The sequence $\mathcal{S}$ defined in Construction 3 is an $(2n, 2R)$-CS of length $k(k+1)$ if $k$ is even and $(k+1)^2$ if $k$ is odd.*

***Proof*** The proof is essentially as the one of Theorem 9. The main difference is that the two sequences $\mathcal{A}$ and $\mathcal{B}$ in Construction 2 is replaced with only one sequence in Theorem 9. The sequence $\mathcal{B}$ is replaced by $\mathcal{A}$ to which a *zero* is concatenated at the end. All the associated shifts between the different positions of the sequence are guaranteed by the different parts. The 0 at the end of each part makes sure that we will move to another shift of $\mathcal{A}$. It does not destroy any $n$-tuples since it can be considered as adding another *zero* to the run of $n-1$ *zeros* in one of the interleaved sequences which does not damage any of the covered words of length $n$. □

***Example 12*** The following $(n, 2)$-CSs were obtained by using Construction 3.

Consider a $(8, 1)$-CS of length 40 with a subsequence of 7 consecutive *zeros* apply Construction 3 to obtain a $(16, 2)$-CS of length $40 \cdot 41 = 1640$.

Consider a $(9, 1)$-CS of length 102 with a subsequence of 8 consecutive *ones* apply Construction 3 to obtain an $(18, 2)$-CS of length $102 \cdot 103 = 10506$.

Consider a $(10, 1)$-CS of length 177 with a subsequence of 9 consecutive *zeros* apply Construction 3 to obtain a $(20, 2)$-CS of length $178 \cdot 178 = 31684$. □

## 6 A construction from primitive polynomials

A construction based on a specific type of primitive polynomials usually does not generate a very short sequence. However, one of our specific bounds (see Table 1) comes from such a sequence. Shift-register sequences and especially those which are produced by a linear function or a modification of such function are very useful for various applications and for constructing of other related combinatorial structures [20, 30]. The same approach is taken in this section. Of special interest are binary sequences generated by a linear function and whose length is $2^n - 1$. These sequences are called M-sequences (for maximal length) and their function is derived from a coefficient of a primitive polynomial.

Let $c(x) = \sum_{i=0}^{n} c_i x^i$, where $c_n = c_0 = 1$ and $c_i \in \{0, 1\}$ for $1 \leq i \leq n-1$, be an irreducible polynomial. Define the following infinite sequence $a_0, a_1, a_2, \ldots$, where

$$a_k = \sum_{i=1}^{n} c_i a_{k-i} \tag{1}$$

with the initial nonzero $n$-tuple $(a_{-n}, a_{-n+1}, \ldots, a_{-1})$. If $c(x)$ is a primitive polynomial, i.e., a polynomial whose roots have order $2^n - 1$, then the sequence $\mathcal{A} = [a_0, a_1, a_2, \ldots]$ is called an M-sequence, its period is period $2^n - 1$, and hence only its first $2^n - 1$ terms are considered. In this cyclic sequence each nonzero $n$-tuple appears exactly once as a window of length $n$. Each root of the primitive polynomial generates the field GF$(2^n)$, i.e., it is a primitive element of the field. The $2^n - 1$ consecutive cyclic shifts of $\mathcal{A}$ can be viewed as another isomorphic representation of the field, These details are well-explained in these two books [20, 30].

Consider now another recursion for a sequence defined by

$$b_k = \sum_{i=1}^{n} c_i b_{k-i} + 1 \tag{2}$$

with the initial nonzero $n$-tuple $(b_{-n}, b_{-n+1}, \ldots, b_{-1})$.

**Lemma 13** *The sum* $\sum_{i=1}^{n} c_i$ *is an even integer.*

***Proof*** This follows from the fact that after $n$ consecutive *ones* we should have a *zero* in such a sequence. Another argument is that if $\sum_{i=1}^{n} c_i$ is an odd integer, then $\sum_{i=0}^{n} c_i = 0$, 1 is a root of $c(x)$ and hence $c(x)$ is not a primitive polynomial. □

**Lemma 14** *The sequence* $\mathcal{B} = (b_0, b_1, b_2, \ldots)$ *is the binary complement of the sequence* $\mathcal{A}$.

***Proof*** Let $\mathbf{a} \triangleq (a_j, a_{j+1}, \ldots, a_{j+n-1})$ be an $n$-tuple in the sequence $\mathcal{A}$. By Eq. (1) we have that

$$a_{j+n} = \sum_{i=1}^{n} c_i a_{j+n-i}.$$

Let $\mathbf{b} \triangleq (b_j, b_{j+1}, \ldots, b_{j+n-1})$ be an $n$-tuple in the sequence $\mathcal{B}$, where $b_i = a_i + 1$, for $j \leq i \leq j + n - 1$, i.e., $\mathbf{b} = \bar{\mathbf{a}}$. Therefore, we have that

$$\begin{aligned} b_{j+n} &= \sum_{i=1}^{n} c_i b_{j+n-i} + 1 = \sum_{i=1}^{n} c_i (a_{j+n-i} + 1) + 1 = \sum_{i=1}^{n} c_i a_{j+n-i} + \sum_{i=1}^{n} c_i + 1 \\ &= \sum_{i=1}^{n} c_i a_{j+n-i} + 1 = a_{j+n} + 1 \end{aligned}.$$

This implies that the sequence $\mathcal{B}$ is the binary complement of the sequence $\mathcal{A}$. □

**Corollary 15** *The recursion of Eq.* (1) *generates the sequence* $\mathcal{A}$ *and the all-zero sequence. The recursion of Eq.* (2) *generates the sequence* $\mathcal{B} = \bar{\mathcal{A}}$ *and the all-one sequence.*

**Lemma 16** *Let* $c(x) = \sum_{i=0}^{n} c_i x^i$ *be a primitive polynomial for which* $c_i = 0$ *for* $1 \leq i \leq 2R + 1$ *and consider the sequences* $\mathcal{A}$, $\mathcal{B}$, *the all-zeros sequence and the all-ones sequence. The code that contains these four sequences is an* $(n + 2R + 1, R)$*-CSC.*

***Proof*** Let $X$ be any given $n$-tuple. Since the first $2R + 1$ $c_i$s (except for $c_0$) are *zeros*, it follows that the last $2R + 1$ elements of $X$ are not influencing the result of the next bit for both recursions and hence the addition of the 1 in the sequence $\mathcal{B}$ implies that the next $2R + 1$ bits after $X$ in $\mathcal{A}$ and $\mathcal{B}$ will be complementing. This implies that $X z_1 z_2 \cdots z_{2R+1}$ is in the ball of radius $R$ either by an $(n + 2R + 1)$-tuple of $\mathcal{A}$ that starts with $X$ or by an $(n + 2R + 1)$-tuple of $\mathcal{B}$ that starts with $X$. □

The number of sequences of the code defined in Lemma 16 is four and they can be efficiently concatenated to one $(n + 2R + 1, R)$-CS.

**Theorem 17** *The four sequences of the* $(n + 2R + 1, R)$*-CSC can be combined to a* $(n + 2R + 1, R)$*-CS of length* $2^{n+1} + 2n + 8R + 2$.

**Table 1** Bounds for the length of the shortest $(n, R)$-CS for $9 \le n \le 20$ and $R = 1, 2, 3$

| $n$ | $R = 1$ | $R = 2$ | $R = 3$ |
|---|---|---|---|
| 9 | 62-93[a] | 20[b] | 12[b] |
| 10 | 107-175[a] | 38[b] | 16[b] |
| 11 | 180-283[a] | 38-111[a] | 20[b] |
| 12 | 342-597[a] | 62-161[a] | 34-40[b] |
| 13 | 598-1172[a] | 97-292[a] | 34-93[a] |
| 14 | 1172-2271[a] | 159-525[a] | 44-239[c] |
| 15 | 2048-3516[e] | 310-907[a] | 70-406[a] |
| 16 | 4096-4462[f] | 512-1640[c] | 115-1036[d] |
| 17 | 7419-17719[a] | 859-5952[d] | 187-1480[d] |
| 18 | 14564-95232[d] | 1702-10506[c] | 316-3720[d] |
| 19 | 26309-176170[g] | 2898-31684[h] | 513-7068[d] |
| 20 | 52618-358400[d] | 5330-31684[c] | 892-13300[d] |

a—Computer search, b—Chung and Cooper [10], c—Construction 3, d—Construction 2, e—from the Hamming code, f—from self-dual sequences, g—from primitive polynomial, h—Theorem 3

***Proof*** The span $n$ M-sequence $\mathcal{A}$ has length $2^n - 1$ and the same length has the sequence $\mathcal{B}$. The sequence $\mathcal{A}$ has a subsequence of $n - 1$ consecutive *zeros* and hence it can be combined with the all-zero sequence to an acyclic sequence of length $2^n + n + 4R + 1$. The same can be done for the sequence $\mathcal{B}$ and the all-one sequence and the two sequences can be combined with no overlaps to an $(n + 2R + 1, R)$-CS of length $2^{n+1} + 2n + 8R + 2$. □

***Remark 18*** The sequences generated in Theorem 17 can be combined with some overlap to reduce its length, but this will make only a very small improvement in the length of the covering sequence. It should be noted the required primitive polynomials exist for relative small $n$ and larger one.

Finally, in Table 1 the current best lower and upper bounds on $\mathcal{L}(n, R)$ are presented, where the lower bounds are either by computer search for very small $n$ or the known lower bound on the smallest size of an $(n, R)$-covering code.

## 7 Folding of a covering sequence into a 2D-sequence

After we have looked at covering sequences and covering sequence codes we continue with a generalization of the one-dimensional framework into two-dimensional arrays. The ultimate goal is to construct an $M \times N$ array $\mathcal{A}$ in which for each $m \times n$ matrix $\mathcal{B}$ there exists an $m \times n$ submatrix $\mathcal{X}$ of $\mathcal{A}$ such that $d(\mathcal{B}, \mathcal{X}) \le R$. Two techniques will be presented in this section. The first one is a folding technique and the second one involves different related shifts of a one-dimensional covering sequence.

For the first technique, folding, we will use the following simple lemma that was already observed in [10].

**Lemma 19** *If there exists an $(n, R)$-CS of length $k$, then there exists an $(n, R)$-CS sequence of length $k + n - 1 + \epsilon$ for any $\epsilon \ge 0$.*

***Proof*** If the $(n, R)$-CS $\mathcal{S}$ starts at a certain position, then we can append to $\mathcal{S}$ the first $n-1$ bits and any $\epsilon$ bits after that to keep it an $(n, R)$-CS. □

**Construction 4** Start with an $(mn, R)$-CS $\mathcal{S} = s_0, s_1, \ldots, s_{k-1}$ of length $k$, where without loss of generality $k$ is divisible by $n$ (see Lemma 19, where $0 \le \epsilon < n-1$). The sequence $\mathcal{S}$ is folded row by row into an $M \times n$ array $\mathcal{A}$, where $M = \frac{k}{n}$. The array is extended into an $M \times (2n-1)$ array $\mathcal{A}'$ by adding to each row of the $M \times n$ array $\mathcal{A}$ the next $n-1$ symbols from $\mathcal{S}$ associated with the given row. In other words, the $j$-th row of the array $\mathcal{A}'$, where $0 \le j \le \frac{k}{n} - 1$, is defined by

$$s_{jn+1}, s_{jn+2}, \ldots, s_{jn+n}, s_{jn+n+1}, \ldots, s_{jn+2n-1}, \tag{3}$$

where the indices are taken modulo $k$. □

Before proving that $\mathcal{A}'$ generated in Construction 4 is an $(m \times n, R)$-C2DS we will prove one property of the array $\mathcal{A}'$.

**Lemma 20** *The array $\mathcal{A}'$ generated in Construction 4 contains the $m \times n$ subarray*

$$\begin{array}{cccc} s_{jn+i} & s_{jn+i+1} & \cdots & s_{jn+i+n-1} \\ s_{(j+1)n+i} & s_{(j+1)n+i+1} & \cdots & s_{(j+1)n+i+n-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{(j+m-1)n+i} & s_{(j+m-1)n+i+1} & \cdots & s_{(j+m-1)n+i+n-1}, \end{array}$$

*where $0 \le i \le n-1$, $0 \le j \le \frac{k}{n} - 1$, and indices are taken modulo $k$.*

***Proof*** This is an immediate consequence of Eq. (3) which defines the $j$-th row of the array $\mathcal{A}'$. □

**Corollary 21** *Each $m \times n$ matrix obtained by folding $mn$ consecutive bits of $\mathcal{S}$, row by row, is contained as an $m \times n$ subarray of $\mathcal{A}'$.*

**Theorem 22** *The array $\mathcal{A}'$ generated in Construction 4 is an $(m \times n, R)$-C2DS.*

***Proof*** Given an $m \times n$ matrix $\mathcal{B}$ we have to show that there exists an $m \times n$ subarray $\mathcal{X}$ of $\mathcal{A}'$ such that $d(\mathcal{B}, \mathcal{X}) \le \mathcal{R}$. Let $X_1, X_2, \ldots, X_m$ be the $m$ consecutive rows of $\mathcal{B}$ and let $\mathcal{T} = X_1 X_2 \cdots X_m$ the sequence of length $mn$ obtained by concatenating them. The sequence $\mathcal{T}$ has length $nm$ and hence there exists a subsequence $\mathcal{Y}$ in $\mathcal{S}$ such that $d(\mathcal{T}, \mathcal{Y}) \le R$. By Corollary 21 the $m \times n$ matrix $\mathcal{Y}'$ obtained by folding $\mathcal{Y}$ into an $m \times n$ matrix is contained in $\mathcal{A}'$. Since $d(\mathcal{T}, \mathcal{Y}) \le R$ it follows that $d(\mathcal{B}, \mathcal{Y}') \le R$ which completes the proof. □

Construction 4 implies the following consequence.

**Theorem 23** *If there exists an $(mn, R)$-CS of length $k$, where $n$ divides $k$, then there exists a $(m \times n, R)$-C2DS of size $M \times N$, where $M = \frac{k}{n}$ and $N = 2n - 1$.*

*If there exists an $(mn, R)$-CS of length $k$, where $n$ does not divide $k$, then there exists an $(m \times n, R)$-C2DS of size $M \times N$, where $N = 2n - 1$ and $\frac{k}{n} \le M \le \lceil \frac{k}{n} \rceil + m$.*

The main disadvantage of Construction 4 is that the width of the array $N = 2n - 1$ is not large enough compared to the width of the window $n$. Next, we will be interested in starting with an $(mn, R)$-CS sequence $\mathcal{S} = [s_0, s_1, s_2, \ldots, s_{k-1}]$ and construct an $M \times N$ $(m \times n, R)$-C2DS for which $M$ will be considerably larger than $m$ and $N$ considerably larger than $n$ and the ratio $\frac{MN}{k}$ should be as small as possible (i.e., with as little redundancy as possible). There are a few ways to construct such an array using the same principles as in Construction 4. The sequence is partitioned into $r$ subsequences which form an $(mn, R)$-CSC. The $r$ sequences should be of the same length $\kappa$ which is divisible by $n$. From each sequence we create an $M \times (2n-1)$ array and we concatenate these $r$ arrays into one $M \times (2nr-r)$ array which is a $(m \times n, R)$-C2DS. The specific details are omitted as they are the same as in Construction 4.

The folding construction is very simple and asymptotically, it yields optimal $(m \times n, R)$-C2DS if the $(mn, R)$-CS is asymptotically optimal. As we mentioned, Proposition 2 can be obtained by using the folding construction, along with the covering sequence whose existence was proved by Vu [53]. The length of this $(mn, R)$-CS is close up to a factor of $\log mn$ from the sphere-covering bound.

It is natural to think that an $(mn, R)$-CS of length $k_1$, will have a smaller length than the area of an $(m \times n, R)$-C2DS with dimension $M \times N$, i.e., $k_1 \leq M \cdot N$. This is indeed the case when folding is applied, $M \cdot N$ is about twice as large as $k_1$. However, we provide one more construction for covering 2D-sequences from covering sequences, whose outcome is quite surprising as it produces smaller arrays (in area) than the length of the ones obtained by folding the related sequence. Moreover, sometimes it yields an array whose area is smaller than the associated best known covering sequence. This second technique is done by considering different shifts of an one-dimensional $(n, R)$-CS. It will be presented in two constructions, where the first one is very simple.

**Construction 5** Let $\mathcal{S}$ be an $(n, R)$-CS of length $k$.

If $k$ is even, then form an $(k+1) \times k$ array whose $i$-th row, $0 \leq i \leq k-1$ is $\mathbf{E}^j S$, where $j = \sum_{\ell=0}^{i} \ell$, $\mathbf{E}^j S$ is a cyclic shift of $\mathcal{S}$ by $j$ positions to the left, i.e.,

$$\mathbf{E}^j[s_0, s_1, \ldots, s_{k-1}] = [s_j, s_{j+1}, \ldots, s_{k-1}, s_0, \ldots, s_{j-1}],$$

and the $k$-th row is the same as the $(k-1)$th row.

If $k$ is odd, then form an $k \times k$ array whose $i$-th row, where $0 \leq i \leq k-1$ is $\mathbf{E}^j S$, $j = \sum_{\ell=0}^{i} \ell$. □

**Theorem 24** *If $n$ is even, then the $(k+1) \times k$ array obtained in Construction 5 is an $(2 \times n, 2R)$-C2DS. If $n$ is odd, then the $k \times k$ array obtained in Construction 5 is an $(2 \times n, 2R)$-C2DS.*

***Proof*** Let $\mathcal{A}$ a $k \times k$ array obtained in Construction 5 and let $\mathcal{B}$ a $2 \times n$ matrix that consists of two sequences (rows) of length $n$, the first one $X$ and the second one $Y$. Each row of $\mathcal{A}$ is the sequence $\mathcal{S}$ at some shift, where the sequence $\mathcal{S}$ of the $i$-th row, $1 \leq i \leq k$, is shifted by $i$ positions to the left compared to the sequence in the $(i-1)$-th row. The 0-th row is taken without shifting.

Assume first that $k$ is even. Since $\mathcal{S}$ is an $(n, R)$-CS, it follows that there exists a subsequence $U$ of length $n$ in $\mathcal{S}$ such the $d(X, U) \leq R$ and a subsequence $V$ of length $n$ in $\mathcal{S}$ such that $d(Y, V) \leq R$. Since $\mathcal{S}$ is an $(n, R)$-CS and all possible shifts are taken between consecutive rows, it follows that in two consecutive rows of $\mathcal{A}$ we have the subsequence $U$ and $V$ one on top of the other, i.e., a $2 \times n$ matrix $\mathcal{W}$ such that $d(\mathcal{W}, \mathcal{B}) \leq 2R$. Thus, the $(k+1) \times k$ array obtained in Construction 5 is an $(2 \times n, 2R)$-C2DS.

When $k$ is odd the proof is similar, but we have to notice that since $\sum_{i=1}^{k} i \equiv 0 \pmod{k}$, it follows that the 0-th row is without any shift compared to the last row and hence one less row is required. □

***Example 25*** Consider the $(6, 1)$-CS of length 12 presented in Appendix A. By applying the defined shifts of Construction 5 on this sequence which is the first row in the array we obtain a $13 \times 12$ $(2 \times 6, 2)$-C2DS whose area is 156. The best associated $(12, 2)$-CS that we find is based on computer search has length 161 (see Appendix B). The generated $13 \times 12$ $(2 \times 6, 2)$-C2DS whose area is 156 which smaller than the related (12, 2)-CS of length 161 that was found by computer search. It is the following array:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

□

***Example 26*** Consider the (7, 1)-CS of length 22 presented in Appendix A. By applying the defined shifts of Construction 5 on this sequence which is the first row in the array, we obtain a $23 \times 22$ $(2 \times 7, 2)$-C2DS whose area 506. The best related (14, 2)-CS that we found is based on computer search has length 525 (see Appendix B). □

Finally, Construction 5 can be generalized as follows.

**Construction 6** Let $\mathcal{S}$ be an $(n, R)$-CS of length $k$ and let $\mathcal{T} = [t_1, t_2, \ldots, t_{k^{m-1}}]$ be a span $m-1$ de Bruijn sequence over $\Sigma_k$. We form the following $k^{m-1} \times k$ array $\mathcal{A}$, where the sequence $\mathcal{S}$ in the $i$-th row of $\mathcal{A}$, $1 \leq i \leq k^{m-1}$ is shifted $t_i$ positions related to the sequence $\mathcal{S}$ in the $(i-1)$-th row, where the 0-th row is considered to be with no shift. □

**Theorem 27** *The $k^{m-1} \times k$ array $\mathcal{A}$ of Construction 6 is an $(m \times n, mR)$-C2DS.*

***Proof*** The proof is very similar to the one of Theorem 24. First note that the sum of all the shifts is zero and hence the virtual 0-th row can be considered to be with no shifts. For any $m \times n$ matrix $\mathcal{B}$ we have to find an $m \times n$ window $\mathcal{X}$ in $\mathcal{A}$ such that $d(\mathcal{B}, \mathcal{W}) \leq mR$. Let $B_1, B_2, \ldots, B_m$ be the $m$ consecutive rows of $\mathcal{B}$. Each $B_i$ has length $n$ and hence there exists a subsequence $X_i$ of length $n$ in $\mathcal{S}$ such that $d(B_i, X_i) \leq R$. Let $(i_1, i_2, \ldots, i_{m-1})$ be the $m-1$ consecutive shifts of $\mathcal{S}$ such that these $m$ copies of $\mathcal{S}$ placed on each other contain $\mathcal{B}$ as a window. Since $(i_1, i_2, \ldots, i_{m-1})$ is an $(m-1)$-tuple over $\mathbb{Z}_k$ and $\mathcal{T}$ is a span $m-1$ de Bruijn sequence over $\mathbb{Z}_k$ which implies that $(i_1, i_2, \ldots, i_{m-1})$ is contained in $\mathcal{T}$, it follows that the associated shifts of $\mathcal{S}$ are contained in $m$ consecutive rows of $\mathcal{A}$. These shifts contain an $m \times n$ window $\mathcal{X}$ whose consecutive rows are $X_1, X_2, \ldots, X_m$. Since $d(X_i, B_i) \leq R$, it follows that $d(\mathcal{B}, \mathcal{X}) \leq mR$.

Thus, the $k^{m-1} \times k$ array $\mathcal{A}$ of Construction 6 is an $(m \times n, mR)$-C2DS. □

## 8 Conclusion and future research

Covering sequences and covering sequence codes which generalize the well-known covering codes, were considered. Some new construction methods for such covering sequences as well as covering sequence codes were presented. In particular, nearly optimal and asymptotically optimal sequences and codes were obtained using the Hamming codes and self-dual sequences. Finally, generalization for covering 2D-sequences was also discussed and related constructions were given.

This area is far from being fully explored and we conclude with the following problems for future research.

(1) The current work and also all the previous papers that considered this topic, have concentrating on the upper bounds of $(n, R)$-CSs. The lower bounds mentioned in this paper are either the ones used for $(n, R)$-covering codes or obtained by computer search. We would like to see improvements in these lower bounds and not just by one or two (something which is not difficult to do).
(2) Many of the upper bounds used in Table 1 are quite weak as Construction 2 is used and we do not have for these parameters a stronger construction like Construction 3. We would like to see some new constructions that will make the table more balanced. Similarly, we want to see upper bounds on the sizes of covering sequence codes.
(3) The $(2^k - 1, 1)$-CS and the $(2^k, 1)$-CS introduced in Sects. 3 and 4 are the best covering sequences obtained for small radii. We would like to have a more precise computation on the length of these sequences. We would also like to see similar sequences for radius 2 and radius 3. The Preparata code used to obtain covering codes for radii 2 and 3 [22, 24] might be the ones to use for this purpose.
(4) We would like to see more constructions as well as lower bounds on the size of covering 2D-sequences.
(5) We would like to see a more comprehensive study on covering sequence codes, mainly $(n, m, R)$-CSCs, as the current work is mainly on covering sequences.
(6) In the same way that covering codes are defined on other metrics, different from the Hamming that was discussed in the paper, covering sequences can be defined for these metrics. For example, it would be interesting to have such short sequences for various poset spaces.

Recently, Rosin [44] developed a heuristic method to search for combinatorial structures. His search found many new such structures including some better covering sequences.

## Appendix A Very small $(n, R)$-CSs used for other bounds

An (8, 1)-CS sequence of length 32 - [00011011111001000001101011100101].

An (8, 1)-CS sequence of length 35 - [00010110110111000010001111011101001].

An (8, 1)-CS sequence of length 37 - [0001101111100100000110101110011100101].

An (8, 1)-CS sequence of length 40 (with 7 consecutive *zeros*) - [0001101111100100000001000001101011100101].

An (8, 2)-CS sequence of length 14 - [00111011010010].

An (9, 2)-CS sequence of length 20 - [00010010001110110111].

## Appendix B Small $(n, R)$-CSs

The following thirteen codewords form a (10, 11, 1)-CSC:

[00001010000], [00101001011], [10100101111], [10110111001], [11011101111],

[11011101111], [01110111100], [11110010011], [11110010000], [11001000101],
[00100011000], [01000110101], [10001101001], [00011010000] .

Their extension by nine bits and their associated overlaps are as follows:

| | | | | | |
|---|---|---|---|---|---|
| 0000101000000000010100 | 7 | 0111011110001110111 1 | 4 | 01000110101010001101 | 8 |
| 00101001011001010010 | 7 | 11110010011111100100 | 9 | 10001101001100011010 | 8 |
| 10100101111101001011 | 4 | 11110010000111100100 | 7 | 00011010000000110100 | 2 |
| 10110111001101101110 | 7 | 11001000101110010001 | 7 | | |
| 11011101111110111011 | 7 | 00100011000001000110 | 8 | | |

The total number of bits in the thirteen sequences is 260 and the total number of overlaps is 85. This yields a (10, 1)-CS of length $260 - 85 = 175$:

[000010100000000010100101100101001011111010010
110111001101101110111111011101111000111101111
001001111110010000111100100010111001000110000
00100011010101000110100110001101000000001101] .

The following thirteen codewords form a (10, 11, 1)-CSC:

[11010111111], [01011110000], [10111100101], [11100110011], [11001100001]
[00110001101], [10001101100], [11011010101], [01101010010], [10101000011],
[10000010010], [00000100000], [00001000101] .

Their extension by nine bits and their associated overlaps are as follows:

| | | | | | |
|---|---|---|---|---|---|
| 11010111111110101111 | 7 | 00110001101001100011 | 6 | 10000010010100000100 | 8 |
| 01011110000010111100 | 8 | 10001101100100011011 | 5 | 00000100000000001000 | 8 |
| 10111100101101111001 | 6 | 11011010101110110101 | 7 | 00001000101000010001 | 1 |
| 11100110011111001100 | 8 | 01101010010011010100 | 7 | | |
| 11001100001110011000 | 7 | 10101000011101010000 | 5 | | |

The total number of bits in the thirteen sequences is 260 and the total number of overlaps is 83. This yields a (10, 1)-CS of length $260 - 83 = 177$ with a subsequence having 10 consecutive *zeros*:

[1101011111111010111100000101111001011011 1100
110011111001100001110011000110100110001101 10
0100011011010101110110101001001101010000 1110
101000000100101000001000000000010001010000 1000] .

The following twenty codewords form a (11, 11, 1)-CSC:

[00111011011], [01110110100], [10110101100], [10101100010], [10001001011],
[10001001010], [01010100111], [01010011010], [01001101110], [10111111110],
[10111111111], [11111111001], [11111000010], [11100001100], [00011011001],
[01100000000], [00000010000], [00001000001], [00000111101], [00011110011] .

Their extension by ten bits and their associated overlaps are as follows:

| | | | | | |
|---|---|---|---|---|---|
| 001110110110011101101 | 9 | 010100110100101001101 | 8 | 000110110010001101100 | 5 |
| 011101101000111011010 | 7 | 010011011100100110111 | 5 | 011000000000110000000 | 6 |
| 101101011001011010110 | 7 | 101111111101011111111 | 10 | 000000100000000001000 | 8 |
| 101011000101010110001 | 5 | 101111111111011111111 | 8 | 000010000010000100000 | 5 |
| 100010010111000100101 | 10 | 111111110011111111100 | 7 | 000001111010000011110 | 8 |
| 100010010101000100101 | 4 | 111110000101111100001 | 8 | 000111100110001111001 | 3 |
| 010101001110101010011 | 8 | 111000011001110000110 | 6 | | |

The total number of bits in the twenty sequences is 420 and the total number of overlaps is 137. This yields a (11, 1)-CS of length $420 - 137 = 283$:

[0011101101100111011010001110110101100101101011000101010 1
1000100101110001001010100010010101001110101010011010010 1
0011011100100110111111110101111111111101111111100111111 11
1000010111111000011001110000110110010001101100000000011 00
000001000000000001000001000010000011110100000111100110001111] .

The following sequences is a (12, 1)-CS of length 597:

[1010110011101101011001111100101100111111110110011111111110110
1111111110000110111111000010100001100001010011110000010100100
0010010100100011110010010001110100011000111010011011011110100
1101100001100110110001001011011000100011001100010001001100000
1000100100010100010010010101011001001010100000100101010001 01
0101101000101011100100010101101010001010111111000101011111 01
1110101111101000110111110100101111011010010111000111101011 10
0011011011110001101110011100110111001010010011100101000001 10
0100100000110000000000001100000111100110000011110000001101111
000000011111100000000101110101000010111010100111011101010 0] .

The following sequences is a (13, 1)-CS of length 1172:

[1011100111110110111001111111001110011111100111100111111 0011
0100101111001101000111000011010001110101100000011101011 0100
1011101011010101000010110101010111001101010101111110000000 10
1111100000100001111100000100010010100001000100010010010001000
1010100000010001010100101100010101001010001101010010100000 11
1100101000000100011010000000100011001110001000110010100110001
1001010110110110010101100111010101011001110110100110011101 1
0010000001110110010011110101100100111110011001001111011111 01
0011110111111110001110111111110100110111111110111110111111110 11
0101011111101101011110111011010111000000110101110001011101 0
1110001000011011100010000000011100010000000000011000000000000
1111110101000011111101001000111111010001111111111010001010 11
1110100010101110101000101011110110000110111101100011001111 0
1100010010110011000100101001000000100101001101001001010011 10

1111000100111011110000101110111100011010101111000110110001
0
1000110110000111000101100001110010011000011100111110000111
0
0100000000111001000110001110010000101011100100001001101101
0
0001001101110000001001101110110010011011101000110110111010
00
1011000110100010110010110000101100101101111111001011011110
1
000101101111010011001011110100111000010101001110000].

The following sequence is a (14, 1)-CS of length 2271:

[1111011110101101111011110101011111011110101001011011110101
0
0100001111010100100011000111100100011000101011100011000101
0
0100001100010100111001000101100111001000101010111001000101
0
1101100100010101000010100010101000011110010101000011010001
0
0010001101000100000110101000100000100000000100000100010000
1
0000010001111111000000100011101110000100011101011000100011
101
0111001000111010101001000111010000100110111010000100000001
0
1000010000001111111011000000111111100101011111111100101000
001
1111001010001100111001010001100100010010001100100000000000
11
0010000000101011001000000010011100100000001000011000000000
10000
1101100000010000110111011110000110111010101010110111010101
00
1011111010101001010011010101001010011001011001010011001111
0
0001001100111100100100100111100100111110011100100111111110
1
0110011111111011100011111111011111101111111011111110001001
0
1111111000110100111111000110100101101000110100101011110110
1
0010101111000100010101111000101110101111000101101011111000
1
0110101100001010110101100011000110101100011001001011100011
0
0100100100001100100100110000010100100110000100001100110000
1
0001111011000010001101111110010001101111101001010101111101
0
0100000101111010010000100100110010000100101001010000100101
01
0100000010010101001110110110101001110111001101001110111010
00
0001110111010001111010111010001110000111010001110011101000
0
0111001110101110011001110101111100001110101111100111110101
1
1110011010100111110011010111010000001101011101001100001011
10
1001101011011101001101111111101001101110011101001101110010
1
1100110111001011110100111001011110011110001011110011110110
0
0111001111011011001001111011011001011101101011001011101100
0
1000101110110000101100110110000101101100010100101101100011
0
0010110110001111100110110001111101100111101111101100111011
0
0110110011101101010110011101100101101011101100101101101101
1
0010110100100110010110100010011010110100010011101010100010
0

$$
\begin{aligned}
&1110001110001001110000000100100111000000010101001100000010101\\
&1010100100101011010100010110000010100010110000000101010110 0\\
&0000010111000100000010111001100011101111001100011111011001 1\\
&0001111000010110001111000000000100111100000000111101110000000\\
&1110011010100001110011011010011110011011010001010011011010 0\\
&0000001101101000011011111101000011011011111000011011011100 0\\
&0001101101110000111101101110000110000000111000011000000110001\\
&0011000000110000101100000011000] .
\end{aligned}
$$

The following six codewords form a (11, 15, 2)-CSC:

$$
\begin{aligned}
&[110011101001001], [111010111001011], [110101110000000],\\
&[010111001000000], [100101000111101], [001010001101100] .
\end{aligned}
$$

Their extension by ten bits and their associated overlaps are as follows:

| | | | | | |
|---|---|---|---|---|---|
| 1100111010010011100111010 | 6 | 1101011100000001101011100 | 8 | 1001010001111011001010001 | 9 |
| 1110101110010111110101110 | 9 | 0101110010000000101110010 | 5 | 0010100011011000010100011 | 2 |

The total number of bits in the six sequences is 150 and the total number of overlaps is 39. This yields a (11, 2)-CS of length $150 - 39 = 111$:

$$
\begin{aligned}
&[110011101001001110011101011100101111101011100000000110101\\
&\ 110010000000101110010100011110110010100011011000010100 0]
\end{aligned}
$$

The following nine codewords form a (12, 13, 2)-CSC:

$$
\begin{aligned}
&[1000101000010], [0100000000100], [1000000001101], [0000110101101], [1010110011110],\\
&[1001110110000], [1000111001001], [1100101111100], [1100101111111] .
\end{aligned}
$$

Their extension by eleven bits and their associated overlaps are as follows:

| | | | | | |
|---|---|---|---|---|---|
| 100010100001010001010000 | 6 | 000011010110100001101011 | 6 | 100011100100110001110010 | 6 |
| 010000000010001000000001 | 10 | 101011001111010101100111 | 6 | 110010111110011001011111 | 11 |
| 100000000110110000000011 | 6 | 100111011000010011101100 | 3 | 110010111111111001011111 | 1 |

The total number of bits in the nine sequences is 216 and the total number of overlaps is 55. This yields a (12, 2)-CS of length $216 - 55 = 161$:

$$
\begin{aligned}
&[10001010000101000101000000000100010000000011011000000000110101\\
&\ 10100001101011001111010101100111011000010011101100011100100 1\\
&\ 100011100101111100110010111111111001011111] .
\end{aligned}
$$

The following sixteen codewords form a (13, 13, 2)-CSC:

$$
\begin{aligned}
&[1111111001011], [1001010011010], [0101001101101], [0011011000110],\\
&[0110001100101], [0011001011011], [0101101000001], [1011010000011],\\
&[0001001001111], [0100111010111], [1011111110111], [1111011100000],\\
&[1110000010001], [1000001000000], [0000000111001], [0111000100001] .
\end{aligned}
$$

Their extension by twelve bits and their associated overlaps are as follows:

| | | | | | |
|---|---|---|---|---|---|
| 1111111001011111111100101 | 6 | 0101101000001010110100000 | 11 | 1110000010001111000001000 | 10 |
| 1001010011010100101001101 | 10 | 1011010000011101101000001 | 4 | 1000001000000100000100000 | 5 |
| 0101001101101010100110110 | 8 | 0001001001111000100100111 | 7 | 0000000111001000000011100 | 6 |
| 0011011000110001101100011 | 8 | 0100111010111010011101011 | 4 | 0111000100001011100010000 | 0 |
| 0110001100101011000110010 | 8 | 1011111110111101111111011 | 7 | | |
| 0011001011011001100101101 | 7 | 1111011100000111101110000 | 7 | | |

The total number of bits in the sixteen sequences is 400 and the total number of overlaps is 108. This yields a (13, 2)-CS of length $400 - 108 = 292$:

[111111100101111111110010100110101001010011011010101001101100
011000110110001100101011000110010110110011001011010000010101
101000001110110100000100100111100010010011101011101001110101
111111011110111111110111000001111011100000100011110000010000
0
010000010000000111001000000011100010000101110001000]

The following sequence is a (14, 2)-CS of length 525:

[001011110100100001011110100111111101110100111111101111100111
111101100011001111101100011011011000100011011011000011111011
011000011100011011000011100100110100011100100111001100010100
111001100101011001001100101011101111111010101110111100000101
101111000000001000111100000000110011010010000110011010100100
11
001101010010010110101010010010110101011110010110101011001101
011101011001101011110111101101011110111010010111100111010010
111011100000010111011100101000101011100101000100000011101000
100000000110000100000000110010001010000110010] .

The following sequence is a (15, 2)-CS of length 907:

[000010111100110000000101111001100100001011100110010000010111
0
110010000001010011000100000101001000000000001010010000011101000
000100001110100101001000111010010100111011101111010011101110
111010011110111011101001000000111011010010000011001000101000
0
011001001010000000110010010101011001111001010101100101111011
1
101100101111010110001111111101011000110100010101100011010011
100101101101001110010011110001111001001111000010010100111100
001001101111111000010011011111010001111101111110100010110111
01
101000101101110011110011000111001111001101110011111100110111
000100010011011100010100101101010001010010110011010001001011
001101101001101100110110100001100011011010000110111001011100
011011100101100000011110010110000011100001111000000111000001
1
001111111000000110011000010000001100110001011000110011000101
01
111100110001010100111010000101010011101011100101001110101111
111011111010111111110110101110101010011010111010101100000011
1
0101011] .

The following five codewords form a (13, 13, 3)-CSC:

[0110111110111], [1111101100010], [0110001101000], [1101000001001], [0100000100000] .

Their extension by twelve bits and their associated overlaps are as follows:

| | | | | | |
|---|---|---|---|---|---|
| 0110111110111011011111011 | 8 | 0110001101000011000110100 | 6 | 0100000100000010000010000 | 1 |
| 1111101100010111110110001 | 7 | 1101000001001110100000100 | 10 | | |

The total number of bits in the five sequences is 125 and the total number of overlaps is 32. This yields a (13, 3)-CS of length 125 − 32 = 93:

[011011111011101101111101100010111110110001101000011000110100
0001001110100000100000001000001000] .

The following ten codewords form a (14, 15, 3)-CSC:

[110011000000010], [011010101001111], [010011101011110], [110101111000100],
[000101100001001], [101101011111101], [111111110001101], [000111011001010],
[001001000101000], [010100001100010] .

Their extension by thirteen bits and their associated overlaps are as follows:

| | | | | | |
|---|---|---|---|---|---|
| 1100110000000101100110000000 | 1 | 0001011000010010001011000010 | 2 | 0010010001010000010010001010 | 5 |
| 0110101010011110110101010011 | 6 | 1011010111111011011010111111 | 6 | 0101000011000100101000011000 | 0 |
| 0100111010111100100111010111 | 8 | 1111111100011011111111100011 | 5 | | |
| 1101011110001001101011110001 | 4 | 0001110110010100001110110010 | 4 | | |

The total number of bits in the ten sequences is 280 and the total number of overlaps is 41. This yields a (14, 3)-CS of length 280 − 41 = 239:

[111001001110101111000100110101111000101100001001000101100001
011010111111011011010111111110001101111111110001110110010100
0011101100100100010100000100100010100001100010010100000011000] .

The following sequence is a (15, 3)-CS of length 406:

[100000010100101100000010100100000010001010010000001000000010110
0101000000101100100011110101100100011111100011101011111110001
1101111101011111101111101011111000110011011111000110001101111
1000110001101100010111011101100010111011011110011111101101111
0011010100000110011010100000111001100100000011100110111000010
1001101110000101000000000000010100000001110100100000001110101
101001111110101101001111100110010101111110011100] .

# Appendix C (15, 1)-CS from the Hamming code of length 15

| | |
|---|---|
| 00001011101010100001011101010 | 1 |
| 00010011001110100010011001110 | 1 |
| 00001010001110100001010001110 | 7 |
| 00011100111111100011100111111 | 8 |
| 00111111111010100111111111010 | 1 |
| 00010011100101100010011100101 | 5 |
| 00101101101110100101101101110 | 1 |
| 00000111100001100000111100001 | 6 |
| 1000010000100001000 | 8 |
| 00001000100010100001000100010 | 9 |
| 00010001001001100010001001001 | 9 |
| 00100100111101100100100111101 | 8 |
| 00111101010110100111101010110 | 1 |
| 00011001111110100011001111110 | 1 |
| 00011011010010100011011010010 | 4 |
| 00101011111110100101011111110 | 1 |
| 00001101111010100001101111010 | 1 |
| 00001110011001100001110011001 | 7 |
| 00110011011010100110011011010 | 1 |
| 00000110101110100000110101110 | 1 |
| 00001110110010100001110110010 | 4 |
| 00101110100110100101110100110 | 1 |
| 00000101100110100000101100110 | 1 |
| 00000011110110100000011110110 | 1 |
| 00000010010010100000010010010 | 8 |
| 10010010010010010 | 7 |
| 00100101011001100100101011001 | 3 |
| 00100101110010100100101110010 | 1 |
| 00000001011010100000001011010 | 10 |
| 00010110100100100010110100100 | 5 |
| 00100111011110100100111011110 | 1 |
| 00011100100110100011100100110 | 8 |
| 00100110111010100100110111010 | 1 |
| 00011010110110100011010110110 | 1 |
| 00001100101100100001100101100 | 2 |
| 00010101101100100010101101100 | 2 |
| 00001010111100100001010111100 | 2 |
| 00011111011100100011111011100 | 2 |
| 00011001001100100011001001100 | 2 |
| 00010011111100100010011111100 | 2 |
| 00000110011100100000110011100 | 2 |
| 00000000100111100000000100111 | 10 |
| 00001001110100100001001110100 | 2 |
| 00000000010101100000000010101 | 10 |
| 00000101010100100000101010100 | 2 |
| 00001101001000100001101001000 | 3 |
| 00001011011000100001011011000 | 3 |
| 00000111111000100000111111000 | 3 |
| 00000001101000100000001101000 | 3 |
| 00000100110000100000100110000 | 4 |
| 00000010100000100000010100000 | 5 |
| 000000000000000 | 10 |
| 00000000001100100000000001100 | 10 |
| 00000011000100100000011000100 | 6 |
| 00010001100010100010001100010 | 5 |
| 00010101000111100010101000111 | 6 |
| 00011111101110100011111101110 | 1 |
| 00010100111010100010100111010 | 8 |
| 00111010111011100111010111011 | 0 |
| 00010010101010100010010101010 | 9 |
| 01010101011101101010101011101 | 0 |
| 00010010110011100010010110011 | 0 |
| 00001111001111100001111001111 | 6 |
| 00111101001111100111101001111 | 6 |
| 00111110110101100111110110101 | 0 |
| 00001011110011100001011110011 | 0 |
| 00001100011110100001100011110 | 8 |
| 00011110010011100011110010011 | 7 |
| 00100111110101100100111110101 | 0 |
| 00010110111101100010110111101 | 0 |
| 00001001101101100001001101101 | 0 |
| 00001001011111100001001011111 | 9 |
| 00101111101001100101111101001 | 0 |
| 00001001000110100001001000110 | 6 |
| 00011010011101100011010011101 | 7 |
| 00111011011111100111011011111 | 0 |
| 00001000111011100001000111011 | 9 |
| 00011101101001100011101101001 | 0 |
| 00010001111011100010001111011 | 9 |
| 00111101111101100111101111101 | 0 |
| 00010101110101100010101110101 | 0 |
| 00001100110101100001100110101 | 8 |
| 00110101010011100110101010011 | 0 |
| 00000111010011100000111010011 | 0 |
| 00010101011110100010101011110 | 8 |
| 01011110111011101011110111011 | 0 |
| 00000110110111100000110110111 | 0 |
| 00010111011001100010111011001 | 0 |
| 00010011010111100010011010111 | 6 |
| 01011111011111101011111011111 | 0 |
| 00001010100101100001010100101 | 8 |
| 1010010100101001010 | 6 |
| 00101011010101100101011010101 | 0 |
| 00000101111111100000101111111 | 0 |
| 00001110101011100001110101011 | 0 |
| 00000110000101100000110000101 | 7 |
| 00001010010111100001010010111 | 7 |
| 00101111011011100101111011011 | 8 |
| 11011011011011011 | 0 |
| 00000101001101100000101001101 | 6 |
| 00110110011011100110110011011 | 0 |
| 00010111110010100010111110010 | 4 |
| 00101011100111100101011100111 | 8 |
| 1110011100111001110 | 6 |
| 00111011101101100111011101101 | 0 |
| 00000100101001100000100101001 | 8 |
| 00101001111001100101001111001 | 0 |
| 00000100011011100000100011011 | 8 |
| 00011011001011100011011001011 | 6 |
| 00101110111111100101110111111 | 0 |
| 00011001100111100011001100111 | 0 |
| 00000011101111100000011101111 | 0 |
| 00000011011101100000011011101 | 0 |
| 00000111001010100000111001010 | 6 |
| 00101010011010100101010011010 | 7 |
| 00110100110111100110100110111 | 8 |
| 00110111111111100110111111111 | 9 |
| 111111111111111 | 0 |
| 00000010111001100000010111001 | 0 |
| 00001101010001100001101010001 | 4 |
| 00010111101011100010111101011 | 0 |
| 00011101011011100011101011011 | 0 |
| 00000010001011100000010001011 | 7 |
| 00010110001111100010110001111 | 7 |
| 00011111110111100011111110111 | 8 |
| 1111011110111101111 | 0 |
| 00000001110001100000001110001 | 4 |
| 00010100100011100010100100011 | 5 |
| 00011010101111100011010101111 | 7 |
| 01011111101101101011111101101 | 0 |
| 00001111010110100001111010110 | 8 |
| 1101011010110101101 | 0 |
| 00010110010110100010110010110 | 7 |
| 00101101110111100101101110111 | 0 |
| 0000000010000111000000001000011 | 6 |
| 00001101100011100001101100011 | 7 |
| 1100011000110001100 | 7 |
| 00011001010101100011001010101 | 8 |
| 01010101101111101010101101111 | 0 |
| 00000000111110100000000111110 | 10 |
| 00001111100100100001111100100 | 5 |
| 00100101101011100100101101011 | 0 |
| 00011011111001100011011111001 | 0 |
| 00001111111101100001111111101 | 0 |

The total number of bits in the one hundred forty-four sequences is 4064 and the total number of overlaps is 548. This yields a (15, 1)-CS of length $4064 - 548 = 3516$.

## Appendix D (16, 1)-CS from self-dual sequences of length 32

| | |
|---|---|
| 1110010011010011000110110010110011100100110100010001101100101110111001001101001 | 11 |
| 0100110100100010101100101101110101001101001001101011001011011001010011010010001 | 11 |
| 1101001000110110001011001100100111010011001101100010110111001001110100100011011 | 9 |
| 1000110110100101011100100101101010000101101001010111101001011010100011011010010 | 9 |
| 0110100101000001100111101011111001100001010000011001011010111110011010010100000 | 9 |
| 0101000000101100101011111111001101010000000011001010111111010011010100000010110 | 8 |
| 0001011010000010111010010111110100010010100000101110110101111101000101101000001 | 7 |
| 1000001111000101001111000011101011000011110001010111110000111010100000111100010 | 5 |
| 0001011110000111111010000111100000010011100001111110110001111000000101111000011 | 10 |
| 1111000011011000000001110010011111111000110110000000111100100111111100001101100 | 8 |
| 0110110010001000100100110111011101101101100010001001001001110111011011001000100 | 8 |
| 0100010000101111101110111101000001000100001010111011101111010100010001000010111 | 12 |
| 0010000101110000110011101000111100110001011100001101111010001111001000010111000 | 13 |
| 1000010111000100011010100011101110010101110001000111101000111011100001011100010 | 12 |
| 0010111000101000110100011101011101101110001010001001000111010111001011100010100 | 10 |
| 1100010100100001001110111101111011000100001000010011101011011110110001010010000 | 8 |
| 1001000010001010011011110111010010010000100010110110111101110101100100001000101 | 7 |
| 1000101110111100011101000100000110001011101111100111010001000011100010111011110 | 8 |
| 1101111000100111001000011101100011011110011001110010000110011000110111100010011 | 7 |
| 0010011001100000110110011001111101100110011000001001100110011111001001100110000 | 10 |
| 1100110000110101001100111100101011001100011101010011001110001010110011000011010 | 9 |
| 0000110101111111111100101000000000000110111111111111100100000000000001101011111 | 10 |
| 1010111111000000010100000011101110101111110001000101000000111111101011111100000 | 10 |
| 1111100000110000000001011100111111111010001100000000011111001111111110000011000 | 10 |
| 0000011000110001111110011100111000000110001100111111100111001100000000110001100 0 | 11 |
| 0110001100001000100111001111001101100011000011001001110011110111011000110000100 | 9 |
| 1100001001101100001111011001001111000010011111000011110110000011110000100110110 | 10 |
| 0100110110110100101100100100101101001101111101001011001000001011010011011011010 | 9 |
| 0110110101001101100100101011001001100101010011011001101010110010011011010100110 | 7 |
| 0100110000001101101100111111001001011100000011011010001111110010010011000000110 | 12 |
| 0110000001100110100111111001100100100000011001101101111110011001011000000110011 | 9 |
| 0001100110011010111001100110010100111001100110101100011001100101000110011001101 | 6 |
| 0011010001110010110010111000110101110100011100101000101110001101001101000111001 | 11 |
| 0100011100111011101110001100010001000111000110111011100011100100010001110011101 | 10 |
| 1110011101011110000110001110000111100111000111100001100010100001110011101011111 | 9 |
| 1101011110101011001010000111010011010111100010110010100001010100110101111010101 | 7 |
| 1010101011111011010101010000001001011101011111011010001010000010010101010111110 1 | 10 |
| 0101111101010010101000001010110111011111010100100010000010101101010111110101001 | 8 |
| 1010100100111001010100101100011010101101001110010101011011000110101010010011100 | 10 |
| 0010011100010010110110001110111100100111000100001101100011101101001001110001001 | 11 |

| | |
|---|---|
| 01110001001000001001111011011111011000010010000010001110110111110111000100100000 | 11 |
| 00010010000101001010110111101011010100100001010011101101111010110001001000010 10 | 9 |
| 10000101011000000111101110011111100001000110000001111010100111111000010101100000 | 11 |
| 01010110000010101010100111010101010101100010101010101001111101010101011000000101 | 10 |
| 11000001010000100001111010111101111000010100001000111110101111011100000101000 01 | 5 |
| 00001010010110111111010110100100000110100101101111100101101001000000010100101101 | 13 |
| 00101001011010001101011010010111001110010110100011000110100101110010100101101 00 | 8 |
| 10110100001001000100111111011011101100000010010001001011110110111011010000100 10 | 13 |
| 11010000100101100110111101101001100100001001011000101111011010011101000010010 11 | 5 |
| 01011011110011111010010000110000010111111100111110100000001100000101101111001 11 | 10 |
| 01111001111011011000001000010010011111011110110110000110000100100111100111101 10 | 9 |
| 01111011001010101000010011010101011010110010101010010100110101010111101100101 01 | 13 |
| 11101100101011010001001101010010111111001010110100000011010100101110110010101 10 | 10 |
| 10010101100011110110101001110001100101011000111001101010011100001001010110001 11 | 7 |
| 10001111011111010111000010000010100011111111110101110000000000010100011110111 110 | 9 |
| 11011111000101110010000011101000110111010001011100100010111010001101111100010 11 | 12 |
| 11111000101111110000000111010000011011100010111110010001110100000111111000101 1111 | 11 |
| 10001011111101110111010000000100010001011110101110111010000101000100010111111 011 | 8 |
| 11111011011011110000010010010000111111110110111100000000100100001111101101101 11 | 10 |
| 01101101110100111001001000111100011011011100001110010010001011000110110111010 01 | 9 |
| 01110100101010111000101101011100011101001010001110001011010101000111010010101 01 | 8 |
| 01010101100111111010101001100000010101111001111110101000011000000101010110011 11 | 12 |
| 10101100111100000101001100001111101011001101000001010011001011111010110011110 00 | 11 |
| 11001111000100000011000011100111110011110001100000110000111011111110011110001 000 | 0 |

The total number of bits in the sixty-four sequences is 5056 and the total number of overlaps is 594. This yields a (16, 1)-CS of length $5056 - 594 = 4462$.

**Acknowledgements** Parts of the paper were presented in IEEE International Symposium on Information Theory and appeared in *Proceedings IEEE Symposium on Information Theory,* Athens, Greece 2024, pp. 1343–1348.

**Author contributions** Y.M.C. initiated the problem and gave directions for the research. He also funded the research. T.E. obtained most of the results presented in the paper. He also wrote the paper. H.T. made all the computer searches and also wrote the latex text for the data of the sequences. He also found the arguments and wrote the proof for the probabilistic method which is mentioned in the paper, but no details are given. V.K.V. worked with H.T. on some of his findings. He also was involved in the folding of the sequences into arrays.

**Funding** Open access funding provided by Technion - Israel Institute of Technology.

**Data availability** No datasets were generated or analysed during the current study.

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

## References

1. Berkowitz R., Kopparty S.: Robust positioning patterns. In: Proceedings of the 27th Annual ACM-SIAM Discrete Algorithms, pp. 1937–1951 (2016).
2. Blaum M., Bruck J.: MDS array codes for correcting criss-cross errors. IEEE Trans. Inform. Theory **46**, 1068–1077 (2000).
3. Blaum M., Bruck J., Vardy A.: Interleaving schemes for multidimensional cluster errors. IEEE Trans. Inform. Theory **44**, 730–743 (1998).
4. Blum A., Jiang T., Li M., Tromp J., Yannakakis M.: Linear approximation of shortest superstrings. J. ACM **41**, 630–647 (1994).
5. Boruchovsky A., Etzion T., Roth R.M.: On nearly perfect covering codes. IEEE Trans. Inform. Theory **71**, 2494–2504 (2025).
6. Breitbach M., Bossert M., Zyablov V., Sidorenko V.: Array codes correcting a two-dimensional cluster of errors. IEEE Trans. Inform. Theory **44**, 2025–2031 (1998).
7. Bruckstein A.M., Etzion T., Giryes R., Gordon N., Holt R.J., Shuldiner D.: Simple and robust binary self-location patterns. IEEE Trans. Inform. Theory **58**, 4884–4889 (2012).
8. Chee Y.M., Etzion T., Ta H., Vu V.K.: On de Bruijn covering sequences and arrays. In: Proceedings IEEE Symposium on Information Theory, Greece, Athens, pp. 1343–1348 (2024).
9. Chee Y.M., Dao D.T., Kiah H.M., Ling S., Wei H.: Robust positioning with low redundancy. SIAM J. Comput. **49**, 284–317 (2020).
10. Chung F., Cooper J.N.: de Bruijn cycles for covering codes. Random Struct. Algorithms **25**, 421–431 (2004).
11. Cohen G.D., Karpovsky M.G., Mattson H.F., Schatz J.R.: Covering radius—survey and recent results. IEEE Trans. Inform. Theory **31**, 328–343 (1985).
12. Cohen G.D., Lobstein A.C., Sloane N.J.A.: Further results on the covering radius of codes. IEEE Trans. Inform. Theory **32**, 680–694 (1986).
13. Cohen G., Honkala I., Litsyn S., Lobstein A.: Covering Codes. North-Holland, Amsterdam (1997).
14. Colbourn C.J.: Combinatorial aspects of covering arrays. Matematiche (Catania) **59**, 125–172 (2004).
15. Colbourn C.J., Keri G., Soriano P.P., Schlage-Putch J.-C.: Covering and radius-covering arrays: constructions and classification. Discret. Math. **158**, 1158–1180 (2010).
16. Dougherty R., Janwa H.: Covering radius computations for binary cyclic codes. Math. Comput. **57**, 415–434 (1991).
17. Downie D., Sloane N.: The covering radius of cyclic codes of length up to 31. IEEE Trans. Inform. Theory **31**, 446–447 (1985).
18. Etzion T.: Pseudo-random and de Bruijn array codes. In: IEEE International Symposium on Information Theory, Athens, Greece, pp. 1742–1747 (2024).
19. Etzion T.: Constructions for perfect maps and pseudorandom arrays. IEEE Trans. Inform. Theory **34**, 1308–1316 (1988).
20. Etzion T.: Sequences and the de Bruijn Graph: Properties, Constructions, and Applications. Elsevier, London/San Diego/Cambridge (2024).
21. Etzion T.: On de Bruijn array codes, part I: nonlinear codes. IEEE Trans. Inform. Theory **71**, 1434–1449 (2025).
22. Etzion T., Greenberg G.: Constructions for perfect mixed codes and other covering codes. IEEE Trans. Inform. Theory **39**, 209–214 (1993).
23. Etzion T., Lempel A.: Construction of de Bruijn sequences of minimal complexity. IEEE Trans. Inform. Theory **30**, 705–709 (1984).
24. Etzion T., Mounits B.: Quasi-perfect codes with small distance. IEEE Trans. Inform. Theory **51**, 3938–3946 (2005).
25. Etzion T., Paterson K.G.: Near optimal single-track gray codes. IEEE Trans. Inform. Theory **42**, 779–789 (1996).
26. Etzion T., Vardy A.: Two-dimensional interleaving schemes with repetitions: constructions and bounds. IEEE Trans. Inform. Theory **48**, 428–457 (2002).
27. Etzion T., Yaakobi E.: Error-correction of multidimensional bursts. IEEE Trans. Inform. Theory **55**, 961–976 (2009).

28. Gallant J., Maier D.D., Astorer J.: On finding minimal length superstrings. J. Comput. Syst. Sci. **20**, 50–58 (1980).
29. Gingeras T., Milazzo J., Sciaky D., Roberts R.: Computer programs for the assembly of DNA sequences. Nucleic Acids Res. **7**, 529–543 (1979).
30. Golomb S.W.: Shift Register Sequences. World Scientific, Singapore (2017).
31. Graham R.L., Sloane N.J.A.: On the covering radius of codes. IEEE Trans. Inform. Theory **31**, 385–401 (1985).
32. Horan V., Stevens B.: Locating patterns in the de Bruijn torus. Discret. Math. **339**, 1274–1282 (2016).
33. Janwa H.: Some new upper bounds on the covering radius of binary linear codes. IEEE Trans. Inform. Theory **35**, 110–112 (1989).
34. Jiang T., Li M.: Approximating shortest superstrings with constraints. Theor. Comput. Sci. **134**, 473–491 (1994).
35. Kaplan H., Shafrir N.: The greedy algorithm for shortest superstrings. Inf. Process. Lett. **93**, 13–17 (2005).
36. Kavut S., Tutdere S.: The covering radii of a class of binary cyclic codes and some bch codes. Des. Codes Cryptogr. **87**, 317–325 (2019).
37. Krivelevich M., Sudakov B., Vu V.H.: Covering codes with improved density. IEEE Trans. Inform. Theory **49**, 1812–1815 (2003).
38. Kumar P.V., Wei V.K.: Minimum distance of logarithmic and fractional partial $m$-sequences. IEEE Trans. Inform. Theory **38**, 1474–1482 (1992).
39. MacWilliams F.J., Sloane N.J.A.: Pseudo-random sequences and arrays. Proc. IEEE **64**, 1715–1729 (1976).
40. Mitchell C.J.: Aperiodic and semi-periodic perfect maps. IEEE Trans. Inform. Theory **41**, 88–95 (1995).
41. Mitchell C.J., Wild P.R.: Constructing orientable sequences. IEEE Trans. Inform. Theory **68**, 4782–4789 (2022).
42. Morano R.A., Ozturk C., Conn R., Dubin S., Zietz S., Nissano J.: Structured light using pseudorandom codes. IEEE Trans. Pattern Anal. Mach. Intell. **20**, 322–327 (1998).
43. Paterson K.G.: Perfect maps. IEEE Trans. Inform. Theory **40**, 743–753 (1994).
44. Rosin C.D.: Using reasoning models to generate search heuristics that solve open instances of combinatorial design problems. https://arxiv.org/abs/2505.23881 (2025).
45. Roth R.M.: Maximum-rank array codes and their application to crisscross error correction. IEEE Trans. Inform. Theory **37**, 328–336 (1991).
46. Schwartz M., Bruck J.: Constrained codes as networks of relations. IEEE Trans. Inform. Theory **54**, 2179–2195 (2008).
47. Sloane N.J.A.: Covering arrays and intersection codes. J. Comb. Des. **1**, 51–63 (1993).
48. Stephen G.: String Searching Algorithms. World Scientific, River Edge, Singapore (1994).
49. Storer J.: Data Compression: Methods and Theory. Computer Science Press Inc., Rockville (1987).
50. Struik R.: Covering codes. Ph.D. Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands (1994).
51. Tal I., Etzion T., Roth R.: On row-by-row coding for 2D constraints. IEEE Trans. Inform. Theory **55**, 3565–3576 (2009).
52. Vazirani V.: Approximation Algorithms. Springer, Berlin (2001).
53. Vu V.: de Bruijn covering codes with arbitrary alphabets. Adv. Appl. Math. **34**, 65–70 (2005).
54. Wang L., Hu S., Shayevitz O.: Quickest sequence phase detection. IEEE Trans. Inform. Theory **63**, 5834–5849 (2017).
55. Wei H.: Nearly optimal robust positioning patterns. IEEE Trans. Inform. Theory **68**, 193–203 (2022).