Des. Codes Cryptogr. (2008) 48:207–229
DOI 10.1007/s10623-008-9202-x

OpenAccess

Sets of disjoint snakes based on a Reed-Muller code and covering the hypercube

A. J. van Zanten · Loeky Haryanto

Received: 22 January 2007 / Revised: 19 October 2007 / Accepted: 29 February 2008 /
Published online: 16 April 2008

Abstract A snake-in-the-box code (or snake) of word length n is a simple circuit in an n-dimensional cube Q_n, with the additional property that any two non-neighboring words in the circuit differ in at least two positions. To construct such snakes a straightforward, non-recursive method is developed based on special linear codes with minimum distance 4. An extension of this method is used for the construction of covers of Q_n consisting of 2^{m-1} vertex-disjoint snakes, for $2^{m-1} < n \leq 2^m$. These covers turn out to have a symmetry group of order 2^m.

Keywords Snake-in-the-box-code · Snake · Gray code · Reed-Muller code · Parallel system · Cover · Invariance group

AMS Classifications 11T71 · 14G50

1 Introduction

This paper is about the construction of certain ordered binary codes called *snake-in-the-box codes*, or briefly *snakes*. In general, a snake in a graph is a simple cycle with no chords. A chord of a cycle $\mathcal{S}$ is an edge of the graph which connects two non-consecutive vertices of $\mathcal{S}$. In this paper, we consider snakes in the *hypercube* Q_n. The vertices of Q_n are all 2^n binary n-tuples (also called binary words of length n), and two vertices (i.e. two binary n-tuples)

Communicated by V. A. Zinoviev.

A. J. van Zanten
Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, P.O. Box 5031, Delft 2600 GA, The Netherlands
e-mail: A.J.vanZanten@twi.tudelft.nl

L. Haryanto (✉)
Department of Mathematics, Hasanuddin University, Kampus Tamalanrea, Makassar 90245, Indonesia
e-mail: Lukih2006@gmail.com

are connected by an edge if and only if they differ in just one position (cf. e.g. [6]). A snake in Q_n is called a snake-in-the-box code.

More precisely, suppose $\mathcal{S} = \mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{L-1}$ is a cyclic list of L words in Q_n. The distance $d(\mathbf{w}_i, \mathbf{w}_j)$ between two words $\mathbf{w}_i$ and $\mathbf{w}_j$ in Q_n is defined to be the Hamming distance of the two words. The list distance $l(\mathbf{w}_i, \mathbf{w}_j)$ between the two words is the minimum number of words in $L$when going from $\mathbf{w}_i$ until $\mathbf{w}_j$, or more precisely

$$l(\mathbf{w}_i, \mathbf{w}_j) := \min\{|i-j|, L-|i-j|\}. \tag{1.1}$$

Then $\mathcal{S}$ is a snake-in-the-box code if for every i, j satisfying $0 \le i, j < L$,

$$d(\mathbf{w}_i, \mathbf{w}_{i+1}) = 1 \tag{1.2}$$

and

$$l(\mathbf{w}_i, \mathbf{w}_j) = 1 \Rightarrow d(\mathbf{w}_i, \mathbf{w}_j) = 1, \tag{1.3}$$

where $\mathbf{w}_L$ is identified with $\mathbf{w}_0$. So, a snake-in-the-box code is a cyclic (closed) list of words of Q_n which satisfies the *nearness condition* (1.2) and the *separability condition* (1.3).

Any cycle in Q_n can be specified by a sequence of integers indicating the bit which changes when going from one word to the next, where the bits in a binary word of length n are labeled by $0, 1, \ldots, n-1$ from left to right. This sequence is called *transition sequence*. Let

$$T = t_1, t_2, \ldots, t_L \tag{1.4}$$

be the transition sequence for the snake $\mathcal{S}$ of *length* (or *range*) L. Then the words $\mathbf{w}_{i-1}$ and $\mathbf{w}_i$ only differ in coordinate t_i, $0 < i \le L$. Since $\mathcal{S}$ is cyclic, we have $\mathbf{w}_L = \mathbf{w}_0$.

We emphasize that the adjective ‘cyclic’ for a snake here roughly means that there is essentially no ‘first’ or ‘last’ word in the snake, i.e. the successor of the ‘last’ word is the ‘first’ word. Otherwise, the resulting snake is called an *open snake*. However, some authors (e.g. Casella and Potter [4], Kochut [10]) use the term ‘snake(-in-the-box code)’ for an open, i.e. non-cyclic snake. In terms of its transition sequence (cf. expressions (2.4), (2.8)), a snake is cyclic if and only if every integer in the transition sequence occurs an even number of times.

The snake $\mathcal{S}$ is called *symmetric* if its transition sequence T is of the form

$$T = t_1, t_2, \ldots, t_K, t_1, t_2, \ldots, t_K, \tag{1.5}$$

which implies that $\mathcal{S}$ is cyclic and $L = 2K$ in (1.4). Kautz [7] calls such a snake *a natural code*. In this case, the second half of the word list of the snake can be obtained from the first half by translating all words in the first half over the vector $\mathbf{w}_K + \mathbf{w}_0$.

Many authors have studied the problem of determining upper and lower bounds for the maximal length $s(n)$ of a snake in Q_n(e.g. [1, 3–5, 10–12, 14–17, 22]), and also how to obtain long snakes (e.g. [14, 18]). At present, the exact value of $s(n)$ has been determined only for six values of n, i.e. $s(2) = 4$, $s(3) = 6$, $s(4) = 8$, $s(5) = 14$, $s(6) = 26$ and $s(7) = 48$. All derived upper bounds show that for $n > 7$, the value of $s(n)$ satisfies $\lambda 2^{n-1} \le s(n) < 2^{n-1}$, with $1 > \lambda \ge 0.60156\ldots$ (cf. [1]).

A generalization of the notion of snake is a *set of* (*vertex-*)*disjoint* snakes. Since for $n \ge 4$ at most half of the number of vertices of Q_n can be incident with a snake, it is a natural question to ask for the minimal number $a(n)$ of disjoint snakes of *equal length* which cover all 2^n vertices of Q_n. Such a set is called a *minimal cover* (by snakes) of Q_n. More generally, we call a set of p disjoint snakes of equal length covering Q_n, a *p-cover* of Q_n.

A problem posed by Erdös, is to decide whether Q_n can be covered with at most l disjoint snakes for some fixed value l, i.e. if there is an integer l such that $a(n) \leq l$, for all $n \geq 2$. Wojciechowski [18] proved that for $l = 16$, the answer is affirmative. Lukito [11] showed that such a cover can always be established with *symmetric* snakes for $l \leq 32$.

Constructions of snakes in Q_n are mostly based on techniques of 'extending' snakes existing in Q_m for some $m < n$. The existence of these 'basic snakes' may have been established in the same way or by other means, e.g. by computer search, or just by accident (e.g. [1]). In this way, the best lower bounds for the length of a snake in Q_n have been derived. Similar techniques are applied in [3, 4, 8] for the construction of snakes, and more generally, for the construction of circuit codes. Also Wojciechowski's approach for proving the existence of covers of Q_n by snakes is not based on a straightforward construction of concrete snakes.

A different approach for the construction of snakes is presented by Paterson and Tuliani [14]. This approach exploits the symmetry properties of *necklaces*. A necklace is an ordered list of the words of a constant-weight binary cyclic code. Here also a computer search is used to get some basis objects.

Our major goal in this paper is to construct snakes in a more straightforward way, i.e. by a non-recursive method. Moreover, we require our method to be extendable for the construction of covers of Q_n by vertex-disjoint snakes, possibly improving the result of Wojciechowski [18] *for certain values of n.*

In this paper we shall apply binary linear algebraic $[n, k, d]$-codes. Such a code is a k-dimensional linear subspace of $GF(2)^n$ with the additional property that any two different vectors (or words) have a Hamming distance at least d (cf. e.g. [13]). Many of these codes appear to have a basis of words of weight d [19].

Our construction starts from a minimum-weight(-d) basis of some *linear algebraic code* $\mathcal{C}$. The weight-d basis vectors are arranged according to a *standard Gray code*, resulting in an ordered cyclic list of the codewords of $\mathcal{C}$, such that each codeword is at Hamming distance d from the previous one [19]. This framework constitutes the skeleton of the snake to be constructed.

In order to obtain the whole snake, we have to change the d bits, going from one codeword of $\mathcal{C}$ to the next one, in such an order that the separability condition (no chords) (1.3) of a snake is satisfied. Several variations and generalizations of this method seem to be possible. We refer to [20] where a similar method was used for the construction of *distance preserving codes* and *covers* of $GF(2)^n$ by such codes.

In this paper, we apply the method above for $d = 4$. Apart from being symmetric due to the applied standard Gray code, our snakes have some additional structure, because of the linearity of the underlying code $\mathcal{C}$. This partial linear structure of the snakes is exploited for the construction of covers of Q_n by symmetric snakes.

The outlines of the paper are as follows. Since our construction method heavily relies on Gray codes, we discuss in Sect. 2 a number of properties of the well-known *standard* or *binary-reflected Gray code* $G(n)$. Another major tool throughout the paper is the binary *Reed-Muller code* $R(m-2, m)$, for $m \geq 3$. Therefore, Sect. 2 contains also a number of relevant properties of Reed-Muller codes. The reason for using this code as well as defining it in terms of EG(m, 2), is that its geometrical structure enables us to satisfy the so called fixed-position property, which is an essential element of our approach.

Section 3 contains the details of our method to construct snakes of length 2^{k+2} in Q_n based on a linear $[n, k, 4]$-code. In Sects. 4 and 5, this general method is applied by making use of $R(m-2, m)$, which is a special minimum-distance-4 code with $n = 2^m$. The order in which bits have to be changed going from one codeword of $R(m-2, m)$ to the next one (see above) will be prescribed by using a complete set of parallel flats in EG$(m, 2)$, the underlying structure of $R(m-2, m)$. Finally, in Sect. 6, the (partial) linear structure of the snakes constructed in the previous section is exploited to obtain covers of Q_n by snakes. More precisely, it is proved that one can construct covers of Q_n by 2^{m-1} symmetric snakes, for any n which satisfies $2^{m-1} < n \leq 2^m$, $m \geq 3$. Such a cover is invariant for a translation group of order 2^m. This invariance property is the subject of Sect. 7.

So, not only do we have now a straightforward method to construct snakes and covers of Q_n by snakes, but these snakes and covers also have some algebraic structure. It turns out that such a cover is completely determined by a special kind of basis of $R(m-2, m)$. Moreover, for the ranges $4 < n \leq 8$ and $8 < n \leq 16$, the number of snakes in our covers of Q_n is less than the upperbound 16 derived by Wojciechowski [18].

As for our notation, we shall stick to the widely used convention to label the bits of a codeword of the standard Gray code $G(n)$ from 1 until n, and from *right to left*, since this is convenient for the index problem and for all properties which are proven by induction. On the other hand, if we are dealing with snake-in-the-box codes in Sects. 3 until 7, we shall label the bits of the codewords from 0 until $n-1$, and from *left to right*, since these codewords originate from a vector space $GF(2)^n$ where the labeling of vector components is similar. Since the standard Gray code $G(k)$ is only used as an auxiliary tool to label the k basis vectors of the underlying $[n, k, 4]$-code, the two different conventions do not interfere.

As was already remarked in the first lines of this Introduction, the vertices of the hypercube Q_n are, by definition, the binary words of length n, or equivalently, the vectors of $GF(2)^n$. When discussing the algebraic structure of the snakes to be constructed, we mostly shall use the term 'vector' to denote these vertices. However, when we want to emphasize the graph-theoretical aspects of our construction, we just call them 'vertices'.

2 Relevant properties of standard Gray codes and Reed-Muller codes

Let Q_n be the n-dimensional cube (*hypercube*), or shortly n-cube. This is the graph with all 2^n binary words of length n as vertices, and all pairs of vertices which differ in precisely one coordinate as edges.

A *cyclic Gray code* is a simple cycle in Q_n. If this cycle is incident with all 2^n vertices of Q_n, one speaks of a *complete* cyclic Gray code, otherwise one calls this Gray code *incomplete*. If not stated explicitly in this paper, the term 'cyclic Gray code' implies completeness. A snake, as defined in Sect. 1, is a Gray code which satisfies the extra condition (1.3), and which for $n > 2$ is always incomplete.

A cyclic *Gray code* can also be defined as a Hamilton cycle in Q_n, i.e. circuit in Q_n containing any of the 2^n vertices precisely once. Equivalently, a cyclic Gray code $G(n)$ is a sequence of n-bit words such that two successive words differ in precisely one position, where we interpret the first word of the sequence as the successor of the last word.

The best known example of such a code is the *binary reflected* or *standard Gray code*. If we define

$$G(1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{2.1}$$

then $G(n)$, $n > 1$, can be defined recursively as the list of ordered rows of a $2^n \times n$ matrix

$$G(n) = \begin{pmatrix} 0 & G(n-1) \\ 1 & G^R(n-1) \end{pmatrix} \tag{2.2}$$

where $G^R(n-1)$ stands for the reversed list of $G(n-1)$, i.e. the ith word of $G^R(n-1)$ is the $(2^{n-1}-1-i)$-th word of $G(n-1)$, $0 \leq i \leq 2^{n-1}-1$.

For some fixed $n \geq 1$, all 2^n codewords of $G(n)$ can also be generated by the *symmetric* (or *non-cyclic*) transition sequence S_n as follows. This non-cyclic transition sequence of $G(n)$ can be defined recursively by

$$S_1 = 1, \quad S_n = S_{n-1}, n, S_{n-1}, \tag{2.3}$$

and for every positive integer $n > 1$. If we write

$$S_n = s_1, s_2, \ldots, s_{2^n-1}, \tag{2.4}$$

then with $\mathbf{g}_0 = \mathbf{0}$, the words in the standard Gray code

$$G(n) = \mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{2^n-1} \tag{2.5}$$

are constructed consecutively such that for every $x \in \{1, 2, \ldots, 2^n-1\}$, the nonzero word $\mathbf{g}_x$ is determined by changing in $\mathbf{g}_{x-1}$ the s_x-th bit from the right.
A useful form of the list $G(n)$ is obtained by partitioning it into 2^{n-i} sublists of size 2^i, starting from $\mathbf{0}$, i.e.

$$G(n) = G_i^0(n), G_i^1(n), \ldots, G_i^{2^{n-i}-1}(n) \tag{2.6}$$

for some i, $1 \leq i \leq n-1$. The non-cyclic lists $G_i^j(n)$ all have the same transition sequence S_i. More explicitly, we can write the transition sequence (2.4) as

$$S_n = S_i, i+1, S_i, i+2, \ldots, S_i, n, S_i, \ldots, i+2, S_i, i+1, S_i. \tag{2.7}$$

This form can be derived by repeatedly applying the second equality of (2.3).

Since $G(n)$ is a *cyclic* Gray code, we also introduce its *complete* transition sequence

$$\overline{S}_n := S_n, n, \tag{2.8}$$

where the integer n indicates the transition from the last word to the first word of $G(n)$.

The Gray code $G(n)$, $n > 1$, and its transition sequence $\overline{S}_n$ satisfy many well-known symmetry relations like (1.5) with $K = 2^{n-1}$. All these relations are easily derived from (2.3). For a review and proofs, we refer to [9, 21]. In addition to (1.5), we shall need a few other properties of $G(n)$ in the next sections. Though these properties are not too well-known, we leave out the proofs, since they are almost as simple as the ones we referred to above. In order to formulate these properties, we next introduce the notion of *contents* for some subsequence of a given transition sequence.

Definition 2.1 Let T be a subsequence of the transition sequence of some cyclic Gray code. Then the contents $c(T)$ of T is the set of those integers which occur an odd number of times in T.

If T stands for the series of transitions when going from $\mathbf{g}$ to $\mathbf{h}$ in the list of code-words, the integers in $c(T)$ indicate the bit positions where $\mathbf{g}$ and $\mathbf{h}$ are different, or equivalently, sup $\mathbf{g}+\mathbf{h} = c(T)$, where sup $\mathbf{v}$ stands for the support of a vector $\mathbf{v}$.

Theorem 2.1 *Let T := i, T', j be a subsequence of the transition sequence $\overline{S}_n$ of the standard Gray code G(n), n > 1.*

(i) if $1 < i < j$ or $1 < j < i$, then $|c(T)|$ is odd and $\min c(T) = i-1$ or $\min c(T) = j-1$, respectively;

(ii) *if $i = j$, then $|c(T)|$ is odd and $\min c(T) \geq i$ for $i < n$, while $c(T) \neq \{i\}$, and $c(T) = \{i-1\}$ for $i = n$;*
(iii) *if $i = 1$, $j > 1$ or $i > 1$, $j = 1$, then $|c(T)|$ is even;*
(iv) *if $T^R = T$, then either $i = j < n$ and $c(T) = \{m\}$ for some $m > i$, or $i = j = n$ and $c(T) = \{n-1\}$.*

Theorem 2.2 *Let $\overline{S}_n$ be the transition sequence of the standard Gray code $G(n)$ and let X be an arbitrary fixed subset of $\{1, 2, \ldots, n\}$, $n \geq 3$, with $|X|$ odd, and $i_0 := min\ X$. If one defines $i := i_0 + 1$ then*

(*i*) *if $1 < i < n$, then for any j with $i < j \leq n$, there exists a subsequence $T = i, T', j$ of $\overline{S}_n$ such that $c(T) = X$;*
(*ii*) *if $1 < i \leq n$, then for any j with $i < j \leq n$, there exists a subsequence $T = j, T', i$ of $\overline{S}_n$ such that $c(T) = X$;*
(*iii*) *if $1 < i \leq n$, then for any j with $1 \leq j \leq i-1$, there exists a subsequence $T = j, T'$, j of $\overline{S}_n$ such that $c(T) = X$;*

For explicit proofs, we refer to [21].

Any of the 2^{n-i} sublists $G_i^j(n)$ in (2.6) can be obtained from any other such sublist by adding a fixed binary vector from $GF(2)^n$ to all its vectors, due to their identical transition sequences S_i in (2.7). In order to formulate this property in a precise way, we introduce vectors $\mathbf{e}_{k,l}$in $GF(2)^n$ which have, by definition, ones at positions k and l from the right and zeros elsewhere, where $k, l \in \{1, 2, \ldots, n\}$, with $k \neq l$.

Theorem 2.3 *The sublists $G_i^j(n)$ of the standard Gray code $G(n)$, $n \geq 2$, are related to each other by the recurrence relation*

$$G_i^j(n) = G_i^{j-1}(n) + \mathbf{e}_{i,i+s_j}, \quad 0 < j \leq 2^{n-i}, \tag{2.9}$$

where the integers s_j are the elements of the transition sequence S_{n-i} of $G(n-i)$.

The second major tool for our method to construct snakes are Reed-Muller codes. The binary r-th order Reed-Muller code $R(r, m)$ is a linear $[n, k, d]$-code with

$$n = 2^m, \quad d = 2^{m-r}, \quad k = \sum_{i=0}^{r} \binom{m}{i}$$

for integers $m \geq 2$ and $0 \leq r \leq m$. For a precise definition we refer to standard textbooks like [13]. Occasionally, the codewords $\mathbf{c}$ of $R(r, m)$ are interpreted as characteristic vectors $\chi(S)$ of certain subsets S of the vector space $GF(2)^m$. Moreover, this vector space is identified with the Euclidean Geometry $EG(m, 2)$ (cf. [13]).

An *r-dimensional subspace* or *r-flat* in $EG(m, 2)$ is by definition an r-dimensional subspace of $GF(2)^m$ or a coset of such a subspace. Within this context, the ones in a codeword $\mathbf{c} = \chi(S)$ correspond to the points (zero-dimensional subspaces) of S.

In this paper we adopt the above interpretation, since the geometric notions appear to be convenient and helpful tools to formulate and to understand our construction of snakes in Q_n. In particular the so-called *parallel systems*, which are complete families of disjoint r-flats covering $EG(m, 2)$, will play a main role in our method. As for our notation, we denote the points of $EG(m, 2)$ by $\mathbf{p}_0, \mathbf{p}_1, \ldots\ldots, \mathbf{p}_{2^m-1}$, where the labeling is such that the index i of $\mathbf{p}_i$, when written as a binary number of length m, is identical to the reversed vector $\mathbf{p}_i^R$ (Remember that we label the components of a vector of $GF(2)^m$ from 0 until $m-1$ from left to right).

This choice of labeling implies, for $i, j \in \{0, 1, \ldots, 2^m - 1\}$, that we can write $\mathbf{p}_i + \mathbf{p}_j = \mathbf{p}_{i \oplus j}$, where we assume that i and j are written as binary numbers of length m, and where the symbol $\oplus$ stands for bitwise addition modulo 2, also known as *Nim addition*. In the next we shall mostly denote the points of $EG(m, 2)$ by their labels represented as decimal numbers. Consequently, a subset $S = \{\mathbf{p}_i, \mathbf{p}_j, \ldots, \mathbf{p}_l\}$ will be represented by $S = \{i, j, \ldots\ldots\ldots, l\}$, and a coset $S + \mathbf{p}_a := \{\mathbf{p}_i + \mathbf{p}_a, \mathbf{p}_j + \mathbf{p}_a, \ldots, \mathbf{p}_l + \mathbf{p}_a\}$ is represented by

$$S \oplus a = \{i \oplus a, j \oplus a, \ldots\ldots\ldots, l \oplus a\}. \tag{2.10}$$

A useful property of the code $R(r, m)$ is that it is spanned by its minimum-weight codewords of weight $d = 2^{m-r}$ [13, Ch. 14], which are the characteristic vectors of $(m - r)$-flats in $EG(m, 2)$.

3 A general method to construct snakes applying an [*n*, *k*, 4]-code

Let $\mathcal{C}$ be some $[n, k, 4]$-code with basis $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k)$ such that $\|\mathbf{b}_i\| = 4$, for $1 \le i \le k$. We assume that these basis vectors are ordered with respect to some, still unspecified, criterion giving rise to the ordered basis $\mathbf{B}$.

We can define an ordered list for the codewords of $\mathcal{C}$ in the following way. Starting from the zeroword $\mathbf{c}_0 = \mathbf{0}$, we define the codewords of $\mathcal{C}$ recursively, by using the integers s_j, $1 \le j < 2^k$, of S_k (cf. (2.4))

$$\mathbf{c}_0 := \mathbf{0}, \mathbf{c}_{i+1} = \mathbf{c}_i + \mathbf{b}_{s_{i+1}}, 0 \le i < 2^k - 1. \tag{3.1}$$

Because of the properties of the Gray code and because of the constant weight 4 of all vectors $\mathbf{b}_i$, the list $\mathcal{C}$ is a complete list of the 2^k codewords $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{2^k-1}$ such that each codeword is at Hamming distance 4 from the previous one (cf. also [19]). Moreover, this last property holds cyclically. In order to arrive at a list of $4 \cdot 2^k$ binary words satisfying the condition that each word differs from the previous one in precisely one bit, we transform $\mathbf{c}_i$ into $\mathbf{c}_{i+1}$ in (3.1) by changing the four bits with labels in sup $\mathbf{b}_{s_{i+1}}$, one after another. This gives rise to intermediate words $\mathbf{w}_i^1$, $\mathbf{w}_i^2$ and $\mathbf{w}_i^3$ between $\mathbf{c}_i$ and $\mathbf{c}_{i+1}$, $0 \le i \le 2^k - 1$.

To specify the order in which the four bits have to be changed, we introduce the notion of *ordered block*. If sup $\mathbf{b}_i = \{i_1, i_2, i_3, i_4\}$, we define $B_i = (i_1, i_2, i_3, i_4)$ where the four integers of sup $\mathbf{b}_i$ are ordered according to some rule. This order is called the *internal order* of the block B_i, $1 \le i \le k$.

In this paper, the internal order of the blocks will be determined by a property which we call the *fixed-position property* and which will be defined in Definition 3.1. The list of blocks $\mathcal{B} := (B_1, B_2, \ldots, B_k)$ is called the *block list* corresponding to $\mathbf{B}$. The order of the blocks in $\mathcal{B}$ is called the *outer order* of the blocks. The sequence

$$\overline{S}_k(\mathcal{B}) = B_1 B_2 B_1 B_3 \ldots B_1 B_2 B_1 B_k B_1 B_2 B_1 B_3 \ldots B_1 B_2 B_1 B_k, \tag{3.2}$$

where the block labels are arranged according to $\overline{S}_k$ (cf. (2.4) and (2.8)), can now be interpreted as a transition sequence of length $4 \cdot 2^n$ for binary words of length n, when the symbols B_i in (3.2) are replaced by the ordered sets (i_1, i_2, i_3, i_4), $1 \le i \le k$.

Applying (3.2), starting from the zeroword $\mathbf{0}$, provides us with the following list of words of length n

$$\mathbf{c}_0 := \mathbf{0}, \mathbf{w}_0^1, \mathbf{w}_0^2, \mathbf{w}_0^3, \mathbf{c}_1, \mathbf{w}_1^1, \mathbf{w}_1^2, \mathbf{w}_1^3, \ldots, \mathbf{c}_{2^k-1}, \mathbf{w}_{2^k-1}^1, \mathbf{w}_{2^k-1}^2, \mathbf{w}_{2^k-1}^3. \tag{3.3}$$

We shall prove that under certain conditions concerning the basis $\mathbf{B}$, the outer order and the internal order of the blocks B_i, $1 \le i \le k$, the 2^{k+2} words of (3.3) constitute a snake.

Not only must the words in (3.3) be different, but they also have to be at distance at least 2 (in cyclic sense) from each other when they are not neighbors.

Definition 3.1 A block list $\mathcal{B} = (B_1, B_2, \ldots, B_k)$ is said to satisfy the fixed-position property, if the internal order of the blocks of $\mathcal{B}$ is such that any integer of the set $\{0, 1, \ldots, n-1\}$ has a fixed position in each block of $\mathcal{B}$ in which it occurs. In this case, the set of integers $\{0, 1, \ldots, n-1\}$ can be partitioned into four subsets I_1, I_2, I_3 and I_4 such that an integer $i \in I_a$, $a \in \{1, 2, 3, 4\}$, can only occur in position a in the blocks $B_1, B_2, \ldots, B_k$.

Lemma 3.1 *Let $\mathcal{C}$ be an $[n, k, 4]$-code with an ordered minimum-weight basis* $\mathbf{B}$= *(*$\mathbf{b}_1$, $\mathbf{b}_2$, $\ldots$, $\mathbf{b}_k$*) such that the corresponding list of blocks $\mathcal{B}$=(B_1, $B_2, \ldots, B_k$) satisfies the fixed-position property. Let* $\mathbf{c} \in \mathcal{C}$ *be some codeword with $W := sup$* $\mathbf{c}$. *Then the parity of $|I_a \cap W|$ is the same for all $a \in \{1, 2, 3, 4\}$.*

Proof The proof follows immediately from the expression of $\mathbf{c}$ w.r.t. the basis $\mathbf{B}$, i.e. $\mathbf{c} = \sum_{l=1}^{k} c_l \mathbf{b}_l$, $c_l \in \{0, 1\}$, and from the fixed-position property. □

We shall consider subsequences of (3.2) of type

$$\mathcal{T} = B_i, \mathcal{T}', B_j, \quad 1 \leq i, j \leq k. \tag{3.4}$$

More in particular, we shall consider the contents $c(\mathcal{T})$ of such subsequences (cf. Definition 2.1). Since subsequence (3.4) consists of complete blocks, its contents is the support of some codeword $\mathbf{c} \in \mathcal{C}$. We therefore define $W := \sup \mathbf{c} = c(\mathcal{T})$. If $i = 1$, $j > 1$ or $i > 1$, $j = 1$, $\mathbf{c}$ is the sum of an even number of basis vectors $\mathbf{b}_l \in \mathbf{B}$, whereas in all other cases, $\mathbf{c}$ is the sum of an odd number of basis vectors (cf. Theorem 2.1). On the other hand, if $\mathbf{c}$ is some codeword of $\mathcal{C}$, it can be written as the sum of a number of basis vectors of basis $\mathbf{B}$, i.e.

$$\mathbf{c} = \sum_{l \in X} \mathbf{b}_l, \tag{3.5}$$

where the index set X is some subset of $\{1, 2, \ldots, k\}$. We define for our convenience,

$$i_0 := \min X. \tag{3.6}$$

Using Theorem 2.1(i), we can write for (3.5) in the case $1 < i < j$ that

$$\mathbf{c} = \mathbf{b}_{i_0} + \sum_{l \geq i_0+1} \mathbf{b}_l \tag{3.7}$$

with $i_0 = i - 1$, and where the summation index l now runs through the index set $X \setminus \{i_0\}$. We shall write $\mathbf{e}_j$, $0 \leq j < n$, for the unit vectors in $GF(2)^n$ with $(\mathbf{e}_j)_i = \delta_{ij}$, where i runs from 0 until $n - 1$ from left to right. So, $B_i = (i_1, i_2, i_3, i_4)$ corresponds to the basis vector

$$\mathbf{b}_i = \mathbf{e}_{i_1} + \mathbf{e}_{i_2} + \mathbf{e}_{i_3} + \mathbf{e}_{i_4}.$$

For our convenience, and because of the role of index $i (= i_0 + 1)$ in (3.7), we shall assume occasionally, when studying sequence (3.4), that $i \leq j$. In order to cover all possibilities, we then also have to take into account sequences $B_j, \mathcal{T}', B_i$. We now prove a necessary and sufficient criterion for a block system or block list $\mathcal{B}$ to generate a snake.

Theorem 3.1 *Let $\mathcal{B} = (B_1, B_2, \ldots, B_k)$ be a block list corresponding to an ordered minimum-weight basis of an $[n, k, 4]$-code $\mathcal{C}$ satisfying the fixed-position property. Then the sequence $\overline{S}_k(\mathcal{B})$ of (3.2) is the transition sequence of a snake of length 2^{k+2} if and only if there is no codeword* $\mathbf{c} \in \mathcal{C}$, *as expressed by (3.7), such that its support W is one of the following sets*

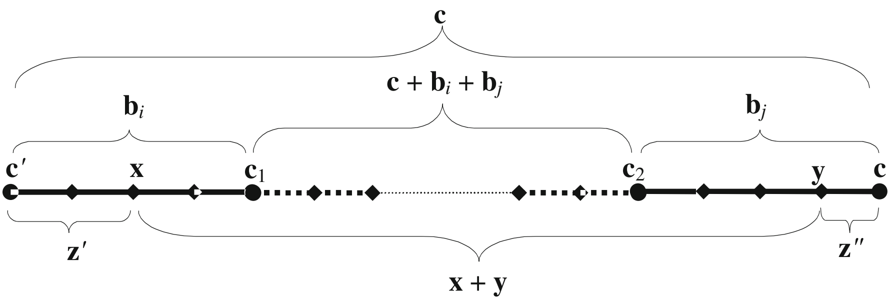

Fig. 1 Partitioning the list of codewords

(i) $W = \{i_1, i_2, p_3, j_4\}$;
(ii) $W = \{i_1, q_2, j_3, j_4\}$;
(iii) $W = \{j_1, j_2, p_3, i_4\}$;
(iv) $W = \{j_1, q_2, i_3, i_4\}$,

for $1 < i < j \leq k$, *and for any p and q with* $p, q \in \{1, 2, \ldots, k\}$.

Proof Let $\mathcal{S}$ be the list of words generated by $\overline{S}_k(\mathcal{B})$ as transition sequence, starting from the zeroword $\mathbf{0}$. Let $\mathbf{x}$ and $\mathbf{y}$ be two words of $\mathcal{S}$. We can write, w.l.o.g., $\mathbf{x} = \mathbf{c}' + \mathbf{z}'$ and $\mathbf{y} = \mathbf{c}'' + \mathbf{z}''$, where $\mathbf{c}', \mathbf{c}'' \in \mathcal{C}$ and where $\mathbf{z}'$ is one of the vectors $\mathbf{0}, \mathbf{e}_{i_1}, \mathbf{e}_{i_1} + \mathbf{e}_{i_2}, \mathbf{e}_{i_1} + \mathbf{e}_{i_2} + \mathbf{e}_{i_3}$, and $\mathbf{z}''$ is one of the vectors $\mathbf{0}, \mathbf{e}_{j_4}, \mathbf{e}_{j_3} + \mathbf{e}_{j_4}, \mathbf{e}_{j_2} + \mathbf{e}_{j_3} + \mathbf{e}_{j_4}$, for some $i, j \in \{1, 2, \ldots, k\}$.

We partition the sequence $\overline{S}_k(\mathcal{B})$ as $\overline{S}_k(\mathcal{B}) = \mathcal{T}'', B_i, \mathcal{T}', B_j, \mathcal{T}'''$ where the sub-sequence $\mathcal{T} := B_i, \mathcal{T}', B_j$ corresponds to the sublist of $\mathcal{S}$ in Fig. 1. If $\mathbf{c} := \mathbf{c}' + \mathbf{c}''$ is the codeword of $\mathcal{C}$ that corresponds to $\mathcal{T}$, we have that $W = \sup \mathbf{c} = c(\mathcal{T})$.

If $\mathcal{T}'$ is empty then either $j > 1, i = 1$ or $i > 1, j = 1$ and it is obvious that $d_{\mathcal{S}}(\mathbf{x}, \mathbf{y}) > 1$ is equivalent to $d(\mathbf{x}, \mathbf{y}) > 1$. In the remaining part of this proof, we assume that $\mathcal{T}'$ is not empty. For similar reasons, we assume that the sublist $\mathcal{T}''', \mathcal{T}''$ is not empty (remember that $\mathcal{S}$ is a circular list). We shall first prove that if the conditions of the theorem hold, the list $\mathcal{S}$ is a snake.

A. Assume $d(\mathbf{x}, \mathbf{y}) = 0$. It follows that $\mathbf{c} = \mathbf{c}' + \mathbf{c}'' = \mathbf{z}' + \mathbf{z}''$. The assumption $\mathcal{T}''', \mathcal{T}'' \neq \emptyset$ implies that $\mathbf{c} \neq \mathbf{0}$. Since $\mathbf{c} \in \mathcal{C}$, the weight $\|\mathbf{c}|$ of $\mathbf{c}$ equals 4 or 6. More in particular, due to the possibilities for $\mathbf{z}'$ and $\mathbf{z}''$, we have the following possible expressions for $\mathbf{c}$.

(a) $\mathbf{c} = \mathbf{e}_{i_1} + \mathbf{e}_{j_2} + \mathbf{e}_{j_3} + \mathbf{e}_{j_4}$;
(b) $\mathbf{c} = \mathbf{e}_{i_1} + \mathbf{e}_{i_2} + \mathbf{e}_{j_3} + \mathbf{e}_{j_4}$;
(c) $\mathbf{c} = \mathbf{e}_{i_1} + \mathbf{e}_{i_2} + \mathbf{e}_{i_3} + \mathbf{e}_{j_4}$;
(d) $\mathbf{c} = \mathbf{e}_{i_1} + \mathbf{e}_{i_2} + \mathbf{e}_{i_3} + \mathbf{e}_{j_2} + \mathbf{e}_{j_3} + \mathbf{e}_{j_4}$.

The cases (*a*), (*c*) and (*d*) do not occur, since then we would have $\|\mathbf{c} + \mathbf{b}_j\| = 2$, $\|\mathbf{c} + \mathbf{b}_i\| = 2$ and $\|\mathbf{c} + \mathbf{b}_i + \mathbf{b}_j\| = 2$, respectively, violating the minimum distance 4 of $\mathcal{C}$. With respect to case (*b*), we remark that $\mathbf{c}$ is the sum of an odd number of basis vectors as a consequence of the fixed-position property, and therefore $i = j$ or $i > 1$, $j > 1$ or both (cf. Theorem 2.1). Now $i = j$ in (*b*) yields $\mathbf{c} = \mathbf{b}_i$ which contradicts Theorem 2.1(*ii*). For $i \neq j$, a word of type (*b*) can not occur because of the conditions of the theorem (take $p = j$ in (*i*) or $q = i$ in (*ii*)).

B. Assume $d(\mathbf{x}, \mathbf{y}) = 1$.

Now, $\mathbf{x} + \mathbf{y} = \mathbf{e}_t$ for some $t \in \{1, 2, \ldots, n\}$, and it follows that $\mathbf{c} = \mathbf{z}' + \mathbf{z}'' + \mathbf{e}_t$. Since $\mathbf{c} \in \mathcal{C}$, we have again that $\|\mathbf{c}\|$ is equal to 4 or 6. The only possibilities for $\mathbf{c}$ with $\|\mathbf{c}\| = 4$ are (cf. (3.4))

(*e*) $\mathbf{c} = \mathbf{e}_t + \mathbf{e}_{j_2} + \mathbf{e}_{j_3} + \mathbf{e}_{j_4}, \quad t \in I_1 \backslash \{i_1, j_1\};$
(*f*) $\mathbf{c} = \mathbf{e}_{i_1} + \mathbf{e}_t + \mathbf{e}_{j_3} + \mathbf{e}_{j_4}, \quad t \in I_2 \backslash \{i_2, j_2\};$
(*g*) $\mathbf{c} = \mathbf{e}_{i_1} + \mathbf{e}_{i_2} + \mathbf{e}_t + \mathbf{e}_{j_4}, \quad t \in I_3 \backslash \{i_3, j_3\};$
(*h*) $\mathbf{c} = \mathbf{e}_{i_1} + \mathbf{e}_{i_2} + \mathbf{e}_{i_3} + \mathbf{e}_t, \quad t \in I_4 \backslash \{i_4, j_4\}.$

Cases (*e*) and (*h*) do not occur, since then we would have $\left\| \mathbf{c} + \mathbf{b}_j \right\| = 2$ and $\|\mathbf{c} + \mathbf{b}_i\| = 2$, respectively, which violates the minimum distance 4 of $\mathcal{C}$. Cases (f) and (g) do not occur because of the conditions of the theorem.

The possibilities for $\mathbf{c}$ with $\|\mathbf{c}\| = 6$ are

(*i*) $\mathbf{c} = \mathbf{e}_{i_1} + \mathbf{e}_{i_2} + \mathbf{e}_t + \mathbf{e}_{j_2} + \mathbf{e}_{j_3} + \mathbf{e}_{j_4}, \quad t \in \mathrm{I}_3 \backslash \{i_3\};$
(*j*) $\mathbf{c} = \mathbf{e}_{i_1} + \mathbf{e}_{i_2} + \mathbf{e}_{i_3} + \mathbf{e}_t + \mathbf{e}_{j_3} + \mathbf{e}_{j_4}, \quad t \in \mathrm{I}_2 \backslash \{j_2\}.$

It will be clear that any choice for t will contradict Lemma 3.1. So, we have proved now the *if* part of the theorem.

Next, we shall prove the *only-if* part. Let $\mathcal{S}$ be a snake. Assume that the conditions of the theorem do not hold. Then there exists a $\mathbf{c} \in \mathcal{C}$ (cf. Eq. 3.7)

$$\mathbf{c} = \sum_{l \in X} \mathbf{b}_l = \mathbf{b}_{i-1} + \sum_{l \geq i} \mathbf{b}_l$$

for some $X \subseteq \{1, 2, \ldots, k\}$, min $X = i - 1$, such that its contents W is equal to one of the sets (*i*)–(*iv*) mentioned in the theorem.

E.g., let $W = \{i_1, i_2, p_3, j_4\}$ with $1 < i < j$. From Theorem 2.2(*i*), we know that the transition sequence $\overline{S}_k$ of the standard Gray code $G(k)$ contains a subsequence $T = i, T', j$ with $c(T) = X$ (remember that the codewords of $\mathcal{C}$ are ordered with respect to the standard Gray code $G(k)$ of length k, which is the dimension of $\mathcal{C}$, whereas n stands for the length of codewords in $\mathcal{C}$, contrary to the role of n in Sect. 2).

It follows that $\overline{S}_k(\mathcal{B})$ contains a subsequence $\mathcal{T} = B_i, \mathcal{T}', B_j$ with

$$c(\mathcal{T}) = W = \{i_1, i_2, p_3, j_4\}, \quad 1 < i < j.$$

If $p_3 = i_3$, we take $\mathbf{z}' = \mathbf{e}_{i_1} + \mathbf{e}_{i_2} + \mathbf{e}_{i_3}$ and $\mathbf{z}'' = \mathbf{e}_{j_4}$ (cf. the beginning of this proof), and we obtain words $\mathbf{x} = \mathbf{c}' + \mathbf{z}'$, $\mathbf{y} = \mathbf{c}'' + \mathbf{z}''$ with mutual distance

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} + \mathbf{y}\| = \left\| \mathbf{c} + \mathbf{z}' + \mathbf{z}'' \right\| = 0.$$

This contradicts our assumption that $\mathcal{S}$ is a snake. Similarly, $p_3 = j_3$ gives $d(\mathbf{x}, \mathbf{y}) = 0$. If $p_3 \neq i_3, j_3$, we take $\mathbf{z}' = \mathbf{e}_{i_1} + \mathbf{e}_{i_2}$ and $\mathbf{z}'' = \mathbf{e}_{j_4}$ giving rise to

$$d(\mathbf{x}, \mathbf{y}) = \left\| \mathbf{e}_{p_3} \right\| = 1,$$

which also contradicts the assumption that $\mathcal{S}$ is a snake. A similar argument can be given in case (*ii*). In cases (*iii*) and (*iv*) we consider the codeword $\mathbf{c}' = \mathbf{c} + \mathbf{b}_i + \mathbf{b}_j$ which can also be expressed in the form of Eq. 3.7 since $i > i - 1$ and $j > i - 1$. Similarly as in cases (*i*) and (*ii*), we now can derive contradictions. Therefore, the conditions of the theorem are necessary as well. □

Example 3.1 We first give a simple example of a snake of range 64 in Q_8, generated by four vectors that are represented by the block list $\mathcal{B} = (B_1, B_2, B_3, B_4)$ with

$$B_1 = (0, 2, 4, 5), \quad B_2 = (1, 2, 4, 6), \quad B_3 = (1, 3, 7, 6), \quad B_4 = (1, 3, 4, 5).$$

These blocks correspond to four independent vectors of weight 4 in $GF(2)^8$, which are the basis vectors of an [8,4,4]-code. It is obvious that they satisfy the fixed-position property. Substituting the blocks in the sequence (3.2) for $k = 4$ gives

$$\overline{S}_4(\mathcal{B}) = B_1B_2B_1B_3B_1B_2B_1B_4B_1B_2B_1B_3B_1B_2B_1B_4.$$

One can easily verify that the codewords $\mathbf{c}$, as expressed by (3.7) cannot be of type (*i*) or (*ii*), nor of type (*iii*) or (*iv*), as formulated in Theorem 3.1. Therefore, the sequence $\overline{S}_4(\mathcal{B})$ is the transition sequence of a snake of range $4 \cdot 2^4 = 2^6$.

In practice, we shall make use of block lists such that the integers $1_1, 2_1, \ldots, t_1$, are all different, while $t_1 = (t+1)_1 = \cdots = k_1$. We shall say that the list is *ordered in standard form* with respect to the integers of I_1, or shortly, that the list is in *standard form*. The integers $1_1, 2_1, \ldots, (t-1)_1$ correspond to *pivot*s in the basis vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{t-1}$, respectively, and can be used to transform the set $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{t-1}\}$ to echelon form. For such a list, we are able to formulate a sufficient criterion for the construction of a snake based on that list, only in terms of the blocks themselves.

Theorem 3.2 *Let $\mathcal{B} = (B_1, B_2, \ldots, B_k)$ be a block list corresponding to an ordered minimum-weight basis of an $[n, k, 4]$-code $\mathcal{C}$, satisfying the fixed-position property. Let furthermore $\mathcal{B}$ be in standard form. Then $\mathcal{B}$ can be transformed into a list $\mathcal{B}'$ corresponding to an equivalent basis of $\mathcal{C}$ such that $\mathcal{B}'$ generates a snake of length 2^{k+2}, if each block B_j of $\mathcal{B}$ with $t \le j < k$, satisfies at least one of the following conditions:*

- *(i)* *the integer j_2 or j_4 does not occur in any of the blocks B_l, $l > j$;*
- *(ii)* *the integer j_3 does not occur in any of the blocks B_l, $l > j$, and $|B_j \cap B_{j+1}| = 1$.*

Proof We shall prove that the conditions of Theorem 3.1 can always be met by transforming $\mathcal{B}$ into an appropriate equivalent block list $\mathcal{B}'$. From the conditions of the theorem we know that the integers $1_1, 2_1, \ldots, (t-1)_1$ are all distinct and that $t_1 = (t+1)_1 = \ldots = k_1$.

Suppose there is a codeword $\mathbf{c} = \mathbf{b}_{i-1} + \sum_{l \ge i} \mathbf{b}_l$ (cf. (3.7)) with sup $\mathbf{c}$ equal to one of the types (*i*)– *iv*) of Theorem 3.1. Then we would have $(i-1)_1 \in$ sup $\mathbf{c}$ for $i \le t$, which contradicts $(i-1)_1 \ne i_1$. For $t < i$, we obtain similar contradictions if B_i satisfies condition (*i*).

Suppose there is a block B_i, $t < i$, which only satisfies condition (*ii*), and let there be a codeword $\mathbf{c} = \mathbf{b}_{i-1} + \sum_{l \ge i} \mathbf{b}_l$ with sup $\mathbf{c}$= $\{t_1, i_2, p_3, j_4\}$, $t < i < j < k$. This cannot happen if $(i-1)_2$ or $(i-1)_4$ does not occur in blocks B_l, $l \ge i$. So, $(i-1)_3$ does not occur in blocks B_l, $l \ge i$, while $(i-1)_2$ and $(i-1)_4$ do. It follows that $p_3 = (i-1)_3$ and $p = i-1$. Now $\mathbf{c} \ne \mathbf{b}_{i-1}$, since $\mathbf{c} = \mathbf{b}_{i-1}$ would imply $(i-1)_2 = i_2$ contradicting $|B_j \cap B_{j+1}| = 1$, for $j = i-1$. We define $\mathbf{b}'_{i-1} := \mathbf{c}$ and replace B_{i-1} in the block list $\mathcal{B}$ by B'_{i-1}, thus defining a new list $\mathcal{B}'$. This list $\mathcal{B}'$ also satisfies the condition that each block contains at least one integer that does not occur in blocks with higher index because $(i-1)_3 \in B'_{i-1}$ and $(i-1)_3 \in B'_l$, $l \ge i$.

The relation $\mathbf{c} = \mathbf{b}_{i-1} + \sum_{l \ge i} \mathbf{b}_l$ must now be written as $\mathbf{c}' := \mathbf{b}_{i-1} = \mathbf{b}'_{i-1} + \sum_{l \ge i} \mathbf{b}_l$ with sup $\mathbf{c}'$=$\{t_1, (i-1)_2, (i-1)_3, (i-1)_4\}$. Since $i_2 \ne (i-1)_2$ and $i_4 \ne (i-1)_4$, this support is not of type (*i*), (*iii*) or (*iv*) of Theorem 3.1, with respect to the new list $\mathbf{B}'$. Neither is it of type (*ii*), because that would imply $\mathbf{c}' = \mathbf{b}_j = \mathbf{b}_{i-1}$, which is false since $j > i$. Similar replacements can be made if $\mathbf{c}$ is of type (*ii*), (*iii*) or (*iv*) of Theorem 3.1. Hence, by carrying out this process for $i = t+1, t+2, \ldots, k$, and anytime if necessary, replacing B_{i-1} by an equivalent independent block B'_{i-1}, we can transform $\mathcal{B}$ into a block list $\mathcal{B}'$ which satisfies the condition of Theorem 3.1. □

Remarks As is clear from the statement as well as from the proof, the conditions of Theorem 3.2 are sufficient. However, they are not necessary for a block list to generate a snake. For a counter example, we refer to the list given in [21, Fig. 3]

The conditions of Theorem 3.2 imply that the code $\mathcal{C}$ has a minimum-weight basis in echelon form with the additional property that, if $j_3 \in I_3$ is the only pivot of $\mathbf{b}_j$, one has

$$\parallel \mathbf{b}_j + \mathbf{b}_{j+1} \parallel = 6.$$

4 Parallel systems in EG(m, 2) and the fixed-position property

Let S be an r-dimensional linear subspace of EG(m, 2). Then S and its cosets are pairwise disjoint. We consider a complete family $\mathcal{P}$ of cosets, i.e.

$$\mathcal{P} := \{S \oplus i_1, S \oplus i_2, \ldots, S \oplus i_l\} \tag{4.1}$$

with $i_1 = 0, i_2, \ldots, i_l \in \{1, 2, \ldots, 2^m - 1\}$ (cf. (2.10)) such that

$$\bigcup_j (S \oplus i_j) = V. \tag{4.2}$$

The disjoint cosets in (4.1) will be called *parallel subspaces* or *parallel flats*. Since the union of cosets contains all points of EG(m, 2), we call P a *parallel system* of flats covering EG(m, 2), or shortly *a cover* of EG(m, 2). Obviously, the number l of parallel r-flats in EG(m, 2) is equal to 2^{m-r}.

In the remaining part of this paper, we shall focus on codes $R(m-2, m)$, $m \geq 3$, which are [n, k, 4]-codes and which are spanned by words of weight 4. We shall prove that for any $m \geq 3$, $R(m-2, m)$ has a weight-4 basis which satisfies the fixed-position property, as defined in Definition 3.1.

We first remark that the sum of all vectors of a linear subspace over $GF(2)$ is equal to $\mathbf{0}$, and hence the Nim sum of their labels equals 0. Since the integers in a coset are obtained from the integers in some linear subspace by addition of a fixed integer, the same property holds for any coset. We now take for S in (4.1) an $(m-2)$-dimensional linear subspace. It follows that $l = 4$, and we write

$$I_j := S \oplus i_j, \quad j \in \{1, 2, 3, 4\}. \tag{4.3}$$

Theorem 4.1 *Let $\mathcal{P}$=\{I_1, I_2, I_3, I_4\} be some parallel system of $(m-2)$-flats covering EG$(m,\ 2)$ and let B be an arbitrary 2-flat. Then the intersections of B with the flats I_i satisfy one of the following relations:*

(i) $|B \cap I_i| = 1$, *for all* $i \in \{1, 2, 3, 4\}$;
(ii) $|B \cap I_i| = |B \cap I_j| = 2$, *for some* $i, j \in \{1, 2, 3, 4\}, i \neq j$;
(iii) $|B \cap I_i| = 4$, *for some* $i \in \{1, 2, 3, 4\}$.

Proof Assume that there is an $i \in \{1, 2, 3, 4\}$ such that $B = \{a, b, c, d\}$ has at least two points - say a and b - in common with I_i. Then it follows that $a \oplus b \in I_1$ and hence, $c \oplus d \in I_1$, since $a \oplus b \oplus c \oplus d = 0$. But then c and d are both in I_j, for some $j \in \{1, 2, 3, 4\}$. If $i = j$, we are in case (iii), and if $i \neq j$ we are in case (ii). If our assumption is false, we are in case (i). □

Theorem 4.2 *For the code $R(m-2, m)$, $m \geq 3$, there exists a minimum-weight basis*

$$\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k)$$

such that the ordered blocks B_i corresponding to sup $\mathbf{b}_i$, $1 \leq i \leq k$, *all satisfy the fixed-position property with respect to any parallel system $\mathcal{P}$ of* $(m-2)$*-flats covering EG*$(m, 2)$.

Proof Let $\mathcal{P}$ be some given parallel system of $(m-2)$-flats. Let B be a 2-flat which is of type (*ii*) (cf. Theorem 4.1) with respect to P, e.g. $B = (a_1, b_1, a_2, b_2)$, where $a_1, b_1 \in I_1$ and $a_2, b_2 \in I_2$. We remark that any 2-flat is uniquely determined by three of its four points and that any of its points is the Nim sum of the others when using the binary representation. We shall say in the next that a block has a *zero-Nim-sum*.

Hence, for any $c_3 \in I_3$, the block (a_1, a_2, c_3, c_4) with $c_4 = a_1 \oplus a_2 \oplus c_3$ stands for a 2-flat. Since $a_1 \oplus a_2 \oplus b_1 \oplus b_2 = 0$, it follows that $b_1 \oplus b_2 \oplus c_3 \oplus c_4 = 0$ and hence (b_1, b_2, c_3, c_4) also represents a 2-flat. Now, we can write (a_1, b_1, a_2, b_2)=(a_1, a_2, c_3, c_4) Δ (b_1, b_2, c_3, c_4), which shows that B can be written as the sum of two blocks satisfying the fixed-position property. Here, the notation $B \Delta B'$ stands for the *symmetric difference* of the blocks B and B' being considered as sets, and it represents the sum of the corresponding vectors.

Similar arguments can be raised for 2-flats of type (*iii*), say for (a_1, b_1, c_1, d_1). For any $e_2, s_2 \in I_2$ and any $f_3 \in I_3$, we can find a $t_3 \in I_3$ such that

$$(a_1, b_1, c_1, d_1) = (a_1, e_2, f_3, g_4)\Delta(b_1, e_2, f_3, h_4)\Delta(c_1, s_2, t_3, g_4)\Delta(d_1, s_2, t_3, h_4),$$

and where the four blocks on the RHS all represent 2-flats. This also illustrates that all blocks representing a 2-flat of type (*iii*) can be written as a sum of blocks satisfying the fixed-position property. From Sect. 2 we know that $R(m-2, m)$ is spanned by its weight-4 vectors, i.e. by some of the 2-flats in EG$(m, 2)$.

From the above considerations it now follows that $R(m-2, m)$ is also spanned by the subset of its weight-4 vectors the corresponding blocks of which satisfy the fixed-position property. □

The following obvious theorem appears to be useful for the construction of minimum-weight bases for $R(m-2, m)$ codes.

Theorem 4.3 *Let* $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k)$ *with* $k = \sum_{i=0}^{m-2} \binom{m}{i}$ *be a set of independent binary vectors of length* $n = 2^m$ *and of weight* 4. *If the corresponding blocks* B_i, $1 \leq i \leq k$, *have zero-Nim-sum, then* $\mathbf{B}$ *generates the Reed-Muller code* $R(m-2, m)$.

Example 4.1 We present the following basis for the code $R(2, 4)$ as a block list $\mathcal{B}$ in standard form.

$$\begin{aligned}
B_1 &= (\underline{3}, 4, 8, 15),\\
B_2 &= (\underline{2}, 4, 8, 14),\\
B_3 &= (\underline{1}, 4, 8, 13),\\
B_4 &= (0, \underline{4}, 8, 12),\\
B_5 &= (0, 7, 8, \underline{15}),\\
B_6 &= (0, 6, \underline{8}, 14),\\
B_7 &= (0, 7, 10, \underline{13}),\\
B_8 &= (0, \underline{6}, 10, 12),\\
B_9 &= (0, 7, 9, \underline{14}),\\
B_{10} &= (0, \underline{5}, 9, 12),\\
B_{11} &= (0, 7, 11, 12).
\end{aligned}$$

The blocks satisfy the fixed-position property with respect to the parallel system $\mathcal{P} = \{I_1, I_2, I_3, I_4\}$, with $I_1 = \{0, 1, 2, 3\}$, $I_2 = \{4, 5, 6, 7\}$, $I_3 = \{8, 9, 10, 11\}$, $I_4 = \{12, 13, 14, 15\}$.

As one can easily verify, each block in the list contains at least one integer which does not occur in blocks with a higher index. For each block we selected such a *pivot* and marked it by an underscore. So, the corresponding basis vectors can be put in echelon form and are independent. Since furthermore all blocks have zero-Nim-sum, the basis B generates the code $R(2, 4)$, which is a $[16, 11, 4]$-code, according to Theorem 4.3. It appears, by applying Theorems 3.1 and 3.2, that the above block list generates a snake in Q_{16} of length 2^{13}. This was verified by a computer program.

In the next section, we shall present a basis for $R(m-2, m)$, for all $m \geq 3$, the corresponding block list of which has a similar structure as the one in Example 4.1.

5 Snakes based on a special basis of $R(m-2, m)$

In EG$(m, 2)$, $m \geq 3$, we introduce the parallel system $\mathcal{P} = \{I_1, I_2, I_3, I_4\}$ consisting of four $(m-2)$-flats defined as

$$\begin{array}{llllll}
I_1 = \{0, & 1, & \ldots, & 2^{m-2}-1 & \}, \\
I_2 = \{2^{m-2}, & 2^{m-2}+1, & \ldots, & 2^{m-1}-1 & \}, \\
I_3 = \{2^{m-1}, & 2^{m-1}+1, & \ldots, & 2^{m-1}+2^{m-2}-1 & \}, \\
I_4 = \{2^{m-1}+2^{m-2}, & 2^{m-1}+2^{m-2}+1, & \ldots, & 2^m-1 & \}
\end{array}$$

with respect to $\mathcal{P}$, we now develop a kind of canonical basis and a corresponding block list $\mathcal{B}(m)$ which we shall call the *canonical block list* of $R(m-2, m)$. This block list $\mathcal{B}(m)$ consists of three sublists $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{B}_3$. Sublist $\mathcal{B}_1$ is defined as

$$\mathcal{B}_1 = \left\{ \begin{array}{lllll}
(\underline{2^{m-2}-1}, & 2^{m-2}, & 2^{m-1}, & 2^m-1 &), \\
(\underline{2^{m-2}-2}, & 2^{m-2}, & 2^{m-1}, & 2^m-2 &), \\
\quad\vdots & \vdots & \vdots & \vdots & \\
(\qquad\underline{1}, & 2^{m-2}, & 2^{m-1}, & 2^{m-1}+2^{m-2}+1 &).
\end{array} \right.$$

The first integer in every block is marked by an underscore, meaning that this integer stands for a pivot in the complete basis to be constructed. Sublist $\mathcal{B}_3$ is built up as follows

$$\mathcal{B}_3^1 = \left\{ \begin{array}{lllll}
(0, & 2^{m-1}-1, & 2^{m-1}, & \underline{2^m-1} &), \\
(0, & 2^{m-1}-2, & \underline{2^{m-1}}, & 2^m-2 &), \\
(0, & 2^{m-1}-1, & 2^{m-1}+2, & \underline{2^m-3} &), \\
(0, & 2^{m-1}-2, & \underline{2^{m-1}+2}, & 2^m-4 &), \\
\vdots & \vdots & \vdots & \vdots & \\
(0, & 2^{m-1}-1, & 2^{m-1}+2^{m-2}-2, & \underline{2^m-2^{m-2}+1} &), \\
(0, & \underline{2^{m-1}-2}, & 2^{m-1}+2^{m-2}-2, & 2^m-2^{m-2} &),
\end{array} \right.$$

$$\mathcal{B}_3^2 = \left\{ \begin{array}{lllll}
(0, & 2^{m-1}-1, & 2^{m-1}+1, & \underline{2^m-2} &), \\
(0, & 2^{m-1}-3, & \underline{2^{m-1}+1}, & 2^m-4 &), \\
(0, & 2^{m-1}-1, & 2^{m-1}+5, & \underline{2^m-6} &), \\
(0, & 2^{m-1}-3, & \underline{2^{m-1}+5}, & 2^m-8 &), \\
\vdots & \vdots & \vdots & \vdots & \\
(0, & 2^{m-1}-1, & 2^{m-1}+2^{m-2}-3, & \underline{2^m-2^{m-2}+2} &), \\
(0, & \underline{2^{m-1}-3}, & 2^{m-1}+2^{m-2}-3, & 2^m-2^{m-2} &),
\end{array} \right.$$

$$\begin{array}{rllll} & \vdots & \vdots & \vdots & \vdots \\ \mathcal{B}_3^{m-2} = \Bigg\{ & (0,\ 2^{m-1}-1, & 2^{m-1}+2^{m-3}-1, & \underline{2^m-2^{m-3}}), \\ & (0,\ \underline{2^{m-1}-2^{m-3}-1}, & 2^{m-1}+2^{m-3}-1, & 2^m-2^{m-2}), \end{array}$$

$$\mathcal{B}_3^{m-1} = (0, 2^{m-1}-1, 2^{m-1}+2^{m-2}-1, 2^m-2^{m-2}).$$

As one can verify, the sublists $\mathcal{B}_3^i$ have size 2^{m-i-1}, $1 \le i \le m-1$. For each block, we mark the integer we selected to play the role of pivot by an underscore. These pivots occur alternately in the third and in the fourth column, except when we are dealing with the last block in $\mathcal{B}_3^i$, $1 \le i < m-1$. For those blocks we selected the integer in the second column. The total number of blocks in $\mathcal{B}_3$ is $\sum_{i=1}^{m-1} 2^{m-i-1} = 2^{m-1}-1$. Finally, sublist $\mathcal{B}_2$ consists of the blocks

$$(0, \underline{i_2}, 2^{m-1}, 2^{m-1}+i_2).$$

Here, i_2 runs through the set $I_2 \backslash I_2'$, where I_2' is the subset of I_2 consisting of those integers that are already present in the blocks of $\mathcal{B}_3$. All integers in $I_2 \backslash I_2'$ are chosen to be pivots. The total number of blocks in $\mathcal{B}_2$ is equal to $|I_2 \backslash I_2'| = 2^{m-2} - m + 1$. Hence, the block list $\mathcal{B}(m) = \mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ contains

$$2^{m-2} - 1 + 2^{m-2} - m + 1 + 2^{m-1} - 1 = 2^m - m - 1$$

blocks. All these blocks have a different pivot, and so they are independent. Furthermore, they all have zero-Nim-sum, as one can easily verify. E.g. block $B_1 (\in \mathcal{B}_1)$ has Nim sum $(2^{m-2}-1) \oplus 2^{m-2} \oplus 2^{m-1} \oplus (2^m-1) = (2^{m-3}+2^{m-4}+\cdots+1) \oplus 2^{m-2} \oplus 2^{m-1} \oplus (2^m-1) = (2^m-1) \oplus (2^m-1) = 0$. So, by Theorem 4.3, the basis corresponding to $\mathcal{B}(m)$ generates $R(m-2, m)$. Moreover, the conditions of Theorem 3.2 are satisfied, and so list $\mathcal{B}(m)$ generates a snake of length 2^{k+2} in Q_n with $k = 2^m - m - 1$ and $n = 2^m$, or is equivalent to such a list.

If $\mathcal{B}(m)$ does not generate a snake immediately but first has to be altered according to the procedure described in the proof of Theorem 3.2, there is a block $B_{i-1} = (0, (i-1)_2, (i-1)_3, (i-1)_4)$ and a codeword $\mathbf{c} = \mathbf{b}_{i-1} + \sum_{l \ge i} \mathbf{b}_l$ with sup $\mathbf{c}$ of type (i), (ii), (iii) or (iv) (cf. Theorem 3.1). Assume sup $\mathbf{c} = \{0, i_2, p_3, j_4\}$, $i < j \le k$ (type (i)). Since $(i-1)_3$ is a pivot in this case, it follows that $p_3 = (i-1)_3$ and $B_{i-1} \in \mathcal{B}_3$. But then all integers l_3 in $\sum_{l \ge i}^{\Delta} B_l$ corresponding to $\sum_{l \ge i} \mathbf{b}_l$ have to occur an even number of times. Because of the structure of the list, the blocks B_l in the above sum occur as pairs of consecutive blocks. But each such pair contains an integer l_4, which is a pivot, so there can be only one such pair.

Let $\mathbf{c} = \mathbf{b}_{i-1} + \mathbf{b}_j + \mathbf{b}_{j+1}$, $j \ge i$. Since we must have $j_3 = (j+1)_3$, $j_2 \ne (j+1)_2$, $j_4 \ne (j+1)_4$, it follows $B_j, B_{j+1} \in \mathcal{B}_3^l$ for some $l \in \{1, 2, \ldots, m-2\}$. Moreover, i_2 and i_4 are never in the same sublist $\mathcal{B}_3^l$. So, sup $\mathbf{c}$ always contains three integers of I_2, or three integers of I_4, and hence, we have a contradiction. Similar contradictions follow when sup $\mathbf{c}$ is of type (ii), (iii) or (iv). Therefore, the list $\mathcal{B}(m)$ itself generates a snake and need not be transformed into an equivalent list. This proves the following theorem.

Theorem 5.1 *The block list $\mathcal{B}(m)$ generates a symmetric snake of length 2^{k+2} in Q_n, where $k = 2^m - m - 1$ and $n = 2^m$, for $m \ge 3$.*

Starting with the basis of $R(m-2, m)$, one can obtain a basis for a code of length $n-1$ and dimension $k-1$, by the process of puncturing (cf. e.g. [13]) with respect to an appropriate coordinate and omitting a relevant block. The reduced list also meets the conditions of Theorem 3.2. Repeating this process, e.g. by puncturing respectively to the pivots of the first y blocks, $1 \le y \le 2^{m-1}-1$, yields the following result.

Corollary 5.2 *For any n which satisfies $2^{m-1} < n \leq 2^m$, one can construct a symmetric snake of length 2^{n-m+1} in Q_n, for every $m \geq 1$.*

For $m \geq 3$, this corollary is an immediate consequence of Theorem 5.1. For $m = 2$, a symmetric snake in Q_3 and Q_4 can easily be constructed. In Q_3, such a snake is generated by the transition sequence 0, 1, 0, 1 and in Q_4, by 0, 1, 2, 3, 0, 1, 2, 3. For $m = 1$, the snake coincides with Q_2 itself.

6 Covers of Q_n by snakes

It is obvious that if we apply a transition sequence of a snake $\mathcal{S}$ in Q_n starting with a word $\mathbf{t}$ different from the zeroword $\mathbf{0}$, we again obtain a snake. Since this snake can also be produced by addition of $\mathbf{t}$ to *all* words of $\mathcal{S}$, we denote it by $\mathcal{S} + \mathbf{t}$, and we call it a *translation* of $\mathcal{S}$ over the vector $\mathbf{t}$. More precisely, if we start with a snake

$$\mathcal{S} = (\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{L-1}), \tag{6.1}$$

then

$$\mathcal{S} + \mathbf{t} = (\mathbf{w}_0 + \mathbf{t}, \mathbf{w}_1 + \mathbf{t}, \ldots, \mathbf{w}_{L-1} + \mathbf{t}). \tag{6.2}$$

An obvious question is whether $\mathcal{S}$ and $\mathcal{S} + \mathbf{t}$ are *disjoint* (i.e. whether they have no common words). In this section we shall study this question with respect to the snakes which are produced by the method of Sect. 3, and which are based on a linear $[n, k, 4]$-code. In the next, the vector $\mathbf{e}_i$ stands for the unit vector with a 'one' on position i and zeros elsewhere, for $0 \leq i \leq n - 1$, just like in Sects. 2 and 3, and we label the positions in a snakeword of length n from 0 until $n - 1$, again from left to right.

Theorem 6.1 *Let $\mathcal{S}$ be the snake produced by the method of Theorem 3.2. The snakes $\mathcal{S}$ and $\mathcal{S} + \mathbf{e}_{p_1} + \mathbf{e}_{q_3}$, and also the snakes $\mathcal{S}$ and $\mathcal{S} + \mathbf{e}_{p_1} + \mathbf{e}_{q_3} + \mathbf{e}_{r_3} + \mathbf{e}_{s_3}$ are disjoint, for all $p, q, r, s \in \{1, 2, \ldots, k\}$, where $p_1 \in I_1$ and $q_3, r_3, s_3 \in I_3$, and where q_3, r_3, s_3 are distinct integers.*

In order to prove e.g. that $\mathcal{S} + \mathbf{e}_{p_1} + \mathbf{e}_{q_3} \cap \mathcal{S} = \emptyset$, one can prove the equivalent statement that there are no vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathcal{S}$ such that $\mathbf{x} + \mathbf{y} = \mathbf{e}_{p_1} + \mathbf{e}_{q_3}$. Writing $\mathbf{x} = \mathbf{c}' + \mathbf{z}'$ and $\mathbf{y} = \mathbf{c}'' + \mathbf{z}''$ with $\mathbf{c}', \mathbf{c}'' \in \mathcal{C}$, one can next eliminate all possibilities for $W := \sup \mathbf{c}' + \mathbf{c}''$ in a completely similar way as in part A of the proof of Theorem 3.1.

Example 6.1 As an example, we consider the snake of Example 4.1, which is a snake of length 2^{13} in Q_{16}. According to Theorem 6.1, we have the following empty intersections:

$$\mathcal{S} \cap \mathcal{S} + \mathbf{e}_0 + \mathbf{e}_8, \mathcal{S} \cap \mathcal{S} + \mathbf{e}_0 + \mathbf{e}_9, \mathcal{S} \cap \mathcal{S} + \mathbf{e}_0 + \mathbf{e}_{10},$$
$$\mathcal{S} \cap \mathcal{S} + \mathbf{e}_0 + \mathbf{e}_{11}, \mathcal{S} \cap \mathcal{S} + \mathbf{e}_0 + \mathbf{e}_i + \mathbf{e}_j + \mathbf{e}_k$$

with $i, j, k \in \{8, 9, 10, 11\}$. Moreover, by computer calculations we found that the following three snakes also have empty mutual intersections: $\mathcal{S} + \mathbf{e}_0 + \mathbf{e}_9$, $\mathcal{S} + \mathbf{e}_0 + \mathbf{e}_{10}$ and $\mathcal{S} + \mathbf{e}_0 + \mathbf{e}_{11}$. Hence, the following set consists of eight snakes which are mutually disjoint:

$$\begin{aligned}\mathcal{C} = \{&\mathcal{S}, \mathcal{S} + \mathbf{e}_0 + \mathbf{e}_9, \mathcal{S} + \mathbf{e}_0 + \mathbf{e}_{10}, \mathcal{S} + \mathbf{e}_0 + \mathbf{e}_{11}, \mathcal{S} + \mathbf{e}_9 + \mathbf{e}_{10}, \mathcal{S} + \mathbf{e}_9 + \mathbf{e}_{11}, \mathcal{S} + \mathbf{e}_{10} + \mathbf{e}_{11},\\ &\mathcal{S} + \mathbf{e}_0 + \mathbf{e}_9 + \mathbf{e}_{10} + \mathbf{e}_{11}\}.\end{aligned}$$

Since $8 \times 2^{13} = 2^{16}$, the set $\mathcal{C}$ is a cover of Q_{16} by eight pairwise disjoint snakes. Since all these snakes are symmetric—they all have the same symmetric transition sequence—we say that $\mathcal{C}$ is a symmetric 8-cover of Q_{16}. We remark that if one replaces one of the vectors $\mathbf{e}_9$, $\mathbf{e}_{10}$ and $\mathbf{e}_{11}$ by $\mathbf{e}_8$, one no longer obtains an 8-cover of Q_{16}. In that case, it turns out that there are four pairs of snakes with a non-empty intersection.

Although all observations in Example 6.1 are verified by computer calculations, one can also prove that the various intersections are empty in a way rather similar to the proofs of Theorems 3.1 and 6.1. However, such proofs are lengthy and tedious, and not very appropriate to be generalized for larger values of n. For this reason, we shall study the problem of disjoint snakes covering a hypercube, from a different point of view. It turns out that the various disjoint snakes of a cover of Q_n constructed by our method, can be connected to each other via a small alteration, such that the result is a complete cyclic Gray code in the very same hypercube Q_n. We shall illustrate this phenomenon by a small example.

Example 6.2 Consider the block list $\mathcal{B} = (\mathcal{B}_1)$ with $\mathcal{B}_1 = (0, 1, 2, 4)$. The vector $\mathbf{b}_1 \in GF(2)^5$ the support of which corresponds to $\mathcal{B}_1$, generates a trivial [5, 1, 4]-code $\mathcal{C}$. The list $\mathcal{B}$ defines the transition sequence

$$\overline{S}_1(\mathcal{B}) = 0, 1, 2, 4, 0, 1, 2, 4$$

which generates a snake $\mathcal{S}$ in Q_5 of length 8. One can verify easily that the four snakes $\mathcal{S}, \mathcal{S} + \mathbf{e}_0 + \mathbf{e}_2, \mathcal{S} + \mathbf{e}_0 + \mathbf{e}_3, \mathcal{S} + \mathbf{e}_2 + \mathbf{e}_3$ are pairwise disjoint and together form a cover of Q_5. We shall construct this cover now in an alternative way.

To this end, we extend the list $\mathcal{B}$ to a list $\mathcal{B}^{ext} = (B_1, B_2, B_3)$, where $B_2 = (0, 1, 3, 4)$ and $B_3 = (0, 1, 0, 4)$. The vector $\mathbf{b}_2$ corresponding to B_2 has a support $\{0, 1, 3, 4\}$, according to our convention in Sect. 3. Furthermore, we assign to B_2 the vector $\mathbf{b}_2$ with support $\{1, 4\}$. It can easily be proved (and verified by the list of words below) that

$$\overline{S}_3(\mathcal{B}^{ext}) = 0, 1, 2, 4, 0, 1, 3, 4, 0, 1, 2, 4, 0, 1, 0, 4, 0, 1, 2, 4, 0, 1, 3, 4, 0, 1, 2, 4, 0, 1, 0, 4$$

is the transition sequence of a cyclic Gray code of wordlength 5. The Gray code itself is presented by the following list of words.

0. 00000 (0)
1. 10000 (1)
2. 11000 (2)
3. 11100 (4)
4. 11101 (0)
5. 01101 (1)
6. 00101 (3)
7. 00111 (4)
8. 00110 (0)
9. 10110 (1)
10. 11110 (2)
11. 11010 (4)
12. 11011 (0)
13. 01011 (1)
14. 00011 (0)
15. 10011 (4)
16. 10010 (0)
17. 00010 (1)
18. 01010 (2)
19. 01110 (4)
20. 01111 (0)
21. 11111 (1)
22. 10111 (3)
23. 10101 (4)
24. 10100 (0)
25. 00100 (1)
26. 01100 (2)
27. 01000 (4)
28. 01001 (0)
29. 11001 (1)
30. 10001 (0)
31. 00001 (4)

It is obvious that its transition sequence can also be written as

$$\overline{S}_3(\mathcal{B}^{ext}) = \mathrm{I}, 3, 4, \mathrm{I}, 0, 4, \mathrm{I}, 3, 4, \mathrm{I}, 0, 4$$

with I = 0, 1, 2, 4, 0, 1. The four subsequences I of length 7 can be interpreted as the transition sequences of four open snakes of length 7. These four snakes are translations of each other, and are separated from each other by the words with indices 7, 15, 23 and 31. By a cyclic permutation of these words to the right, we obtain (closed) snakes of length 8

$$\mathcal{S}^{(4)}, \quad \mathcal{S}^{(4)} + \mathbf{b}_1 + \mathbf{b}_2, \quad \mathcal{S}^{(4)} + \mathbf{b}_2 + \mathbf{b}_3, \quad \mathcal{S}^{(4)} + \mathbf{b}_1 + \mathbf{b}_3,$$

which also are translations of each other. Since $\mathbf{b}_1 + \mathbf{b}_2 = \mathbf{e}_2 + \mathbf{e}_3$, $\mathbf{b}_2 + \mathbf{b}_3 = \mathbf{e}_0 + \mathbf{e}_3$ and $\mathbf{b}_1 + \mathbf{b}_3 = \mathbf{e}_0 + \mathbf{e}_2$, we have a 4-cover of Q_5, which is similar to the 8-cover of Q_{16} mentioned in Example 6.1.

In general, we shall extend a block list $\mathcal{B}$ in standard order which corresponds to the basis of an $[n, k, 4]$-code, and which satisfies the fixed-position property, with a additional *independent* blocks $B_{k+1}, B_{k+2}, \ldots, B_{k+a}$, defined as

$$B_l = (k_1, k_2, l_3, k_4), \tag{6.3}$$

where $l_3 \in I_3$ for $k+1 \leq l \leq k+a-1$ and $l_3 = k_1$ for $l = k+a$.

The blocks of (6.3) correspond to vectors $\mathbf{b}_l \in (GF(2))^n$ with sup $\mathbf{b}_l = \{k_1, k_2, l_3, k_4\}$, for $k+1 \leq l \leq k+a-1$, whereas we let B_{k+a} correspond to a vector $\mathbf{b}_{k+a}$ of weight 2 with sup $\mathbf{b}_{k+a} = \{k_2, k_4\}$. So, the new block list $\mathcal{B}^{ext}$ consists of $k+a$ independent blocks, and is also in standard form since $k_1 = (k+1)_1 = (k+2)_1 = \cdots = (k+a)_1$. However, it no longer satisfies the fixed-position property because of block B_{k+a}, and neither do the new blocks have a place in the underlying Euclidean Geometry of the original list $\mathcal{B}$, since the Nim sum of the integers in a block is unequal to 0. It is obvious that $\mathcal{B}^{ext}$ generates an $[n, k+a, 2]$-code, which we shall call $\mathcal{C}^{ext}$ since it contains $\mathcal{C}$ as a subcode.

The transition sequence $\overline{S}_{k+a}(\mathcal{B}^{ext})$ obtained by substituting the blocks of $\mathcal{B}^{ext}$ in (3.2) will, in general, not define a snake. This is because the minimum distance 2 of $\mathcal{C}^{ext}$ prevents us from proving the separability condition of a snake like we did in part B of the proof of Theorem 3.1. However, we can rather easily prove the validity of the nearness condition by assuming one additional (weak) condition for the blocks, and next slightly generalizing part A of that proof.

Theorem 6.2 *Let $\mathcal{B} = (B_1, B_2, \ldots, B_k)$ be a block list in standard order corresponding to an [n, k, 4]-code satisfying the fixed-position property. Let $\mathcal{B}^{ext} = (B_1, B_2, \ldots, B_k, B_{k+1}, \ldots, B_{k+a})$ be the extended block list such that $\mathcal{B}_{k+1}, \ldots, \mathcal{B}_{k+a}$ are defined by* (6.3). *Then $\overline{S}_{k+a}(\mathcal{B}^{ext})$ is the transition sequence of a complete or incomplete Gray code, if each $\mathcal{B}_i$, $i \in \{1, 2, \ldots, k+a\}$, contains at least one integer that does not occur in any of the blocks $\mathcal{B}_l$, $l > i$.*

Proof We shall prove that all 2^{k+a+2} words of the list $\mathcal{S}^{ext}$ generated by $\overline{S}_{k+a}(\mathcal{B}^{ext})$, starting from the zeroword, are different. The arguments are similar to those used in the proof of part A of Theorem 3.1. Take two words $\mathbf{x}$ and $\mathbf{y}$ from this list $\mathcal{S}^{ext}$. Just like in the proof of the above mentioned theorem, we can write w.l.o.g., $\mathbf{x} = \mathbf{c}' + \mathbf{z}'$ and $\mathbf{y} = \mathbf{c}'' + \mathbf{z}''$, where $\mathbf{c}'$, $\mathbf{c}'' \in \mathcal{C}^{ext}$, and where $\mathbf{z}'$ is one of the vectors $\mathbf{0}$, $\mathbf{e}_{i_1}$, $\mathbf{e}_{i_1} + \mathbf{e}_{i_2}$, $\mathbf{e}_{i_1} + \mathbf{e}_{i_2} + \mathbf{e}_{i_3}$ and $\mathbf{z}''$ is one of the vectors $\mathbf{0}$, $\mathbf{e}_{j_4}$, $\mathbf{e}_{j_3} + \mathbf{e}_{j_4}$, $\mathbf{e}_{j_2} + \mathbf{e}_{j_3} + \mathbf{e}_{j_4}$, $1 \leq i, j \leq k+a$, For a picture of the sublist of $\mathcal{S}^{ext}$ between $\mathbf{x}$ and $\mathbf{y}$, we refer to Fig. 1. Furthermore, we define $\mathbf{c} := \mathbf{c}' + \mathbf{c}''$ and we partition the transition sequence as $\overline{S}_{k+a}(\mathcal{B}^{ext}) = \mathcal{T}'', B_i, \mathcal{T}', B_j, \mathcal{T}'''$ with sup $\mathbf{c} = c(B_i, \mathcal{T}', B_j)$ and with non-empty subsequences $\mathcal{T}'$ and $\mathcal{T}'''$, $\mathcal{T}''$.

Assume that $d(\mathbf{x}, \mathbf{y}) = 0$, or equivalently that $\mathbf{c} := \mathbf{z}' + \mathbf{z}''$. The weight $\|\mathbf{c}\|$ of $\mathbf{c}$ can now be equal to 2, 4 or 6, since $\mathbf{c} \in \mathcal{C}^{ext}$. It follows from the form of the blocks in $\mathcal{B}^{ext}$ that the codewords of $\mathcal{C}^{ext}$ of weight 2 have a support that is either of type $\{p_b, q_b\}$, $b \in \{1, 2, 3, 4\}$, or of type $\{p_1, q_3\}$, or of type $\{p_2, q_4\}$, with $p, q \in \{1, 2, \ldots, k+a\}$. From the possible expressions of $\mathbf{z}'$ and $\mathbf{z}''$ and from (6.3), we may conclude that the only possibility to satisfy $\mathbf{c} = \mathbf{z}' + \mathbf{z}''$ is sup $\mathbf{c} = \{i_2, j_4\}$ with $i = k+a$ or $j = k+a$. Since such a $\mathbf{c}$ satisfies $|\text{sup } \mathbf{c} \cap I_2| = 1$, it is the sum of an odd number of basis vectors, and since $|\text{sup } \mathbf{c} \cap I_1| = 0$, this sum contains $\mathbf{b}_{k+a}$.

If $i = j = k+a$, it follows from Theorem 2.1(*ii*) that $\mathbf{c} = \mathbf{b}_{k+a-1}$ which contradicts the previous conclusion. Suppose $i < j = k+a$. Then $\mathbf{c} = \mathbf{b}_{i-1} + \sum_{l \geq i} \mathbf{b}_l$, where l runs through an index set of even size according to Theorem 2.1(*i*). From the condition of the theorem we know that B_{i-1} contains at least one integer which does not occur in blocks $B_l, l \geq i$. This cannot be an integer from I_1 or I_3, since such an integer does not occur in sup $\mathbf{c}$. But if $(i-1)_2$ does not occur anymore, then it follows that $(i-1)_2 = (k+a)_2 (= k_2)$, which is also a contradiction. For a similar reason $(i-1)_4$ cannot occur in blocks $B_l, l \geq i$. Assuming $j < i = k+a$ gives rise to similar contradictions. So, we may conclude that $\|\mathbf{c}\|$ is either equal to 4 or to 6.

Due to the possible expressions for $\mathbf{z}'$ and $\mathbf{z}''$ and the general form of the blocks of $\mathcal{B}^{ext}$, the only possible vectors $\mathbf{c}$ of weight 4 or 6 are of type (a), (b), (c) or (d) as defined in the proof of Theorem 3.1 with $1 \leq i, j \leq k+a$, but with $i_3 \neq (k+a)_3$ and $j_3 \neq (k+a)_3$. Vectors $\mathbf{c}$ of types (a), (c) and (d) can be eliminated by reducing these cases to the previous case of weight 2. Vectors $\mathbf{c}$ of type (b) can be dealt with in the same way as the vectors of weight 2 by using the condition that each block contains an integer which does not occur in blocks with a higher index. We conclude that $d(\mathbf{x}, \mathbf{y}) \neq 0$, and hence, $\mathcal{B}^{ext}$ generates a Gray code with 2^{k+a+2} words. This Gray code is complete if $a = n - k - 2$ and incomplete if $a < n - k - 2$. □

Theorem 6.3 *Let $\mathcal{B} = (B_1, B_2, \ldots, B_k)$ and $\mathcal{B}^{ext} = (B_1, B_2, \ldots, B_k, B_{k+1}, \ldots, B_{k+a})$, $a = n - k - 2$, be two block lists that satisfy the conditions of Theorem 6.2, and let $\mathcal{B}$ generate a snake of size 2^{k+2}. Let furthermore $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{2^a-1}$ be the 2^a words in the word list $\mathcal{S}^{ext}$ generated by $\mathcal{B}^{ext}$ at the positions $2^{k+2}-1, 2 \cdot 2^{k+2}-1, 3 \cdot 2^{k+2}-1, \ldots, 2^a \cdot 2^{k+2}-1$. Then one obtains a 2^a-cover of Q_n by symmetric snakes, by carrying out the cyclic permutation $(\mathbf{u}_0\mathbf{u}_1 \ldots \mathbf{u}_{2^a-1})$ in $\mathcal{S}^{ext}$.*

Proof We partition $\overline{S}_{k+a}$ $(\mathcal{B}^{ext})$, starting from its first element into 2^a subsequences of size 2^{k+2}. From the expression (2.6) with $i = n - a$, and from the specific form of the blocks $B_k, B_{k+1}, \ldots, \mathcal{B}_{k+a}$, which differ only in their third element, it follows that these 2^a subsequences are all equal to $\overline{S}_k(\mathcal{B})$, apart from their last but one element (cf. Example (6.2)). Hence, they are the transition sequences of 2^a open snakes of length $2^{k+2} - 1$, separated by words $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{2^a-1}$. The open snakes can be obtained from each other by a translation over some fixed vector, as a consequence of Theorem 2.3. These translation vectors are the 2^a linear combinations of the vectors $\mathbf{s}_1 = \mathbf{b}_k + \mathbf{b}_{k+1}, \mathbf{s}_2 = \mathbf{b}_k + \mathbf{b}_{k+2}, \ldots, \mathbf{s}_a = \mathbf{b}_k + \mathbf{b}_{k+a}$,

More precisely, let

$$\mathcal{S}^0, \mathbf{u}_0, \quad \mathcal{S}^1, \mathbf{u}_1, \quad \ldots, \quad \mathcal{S}^{a-1}, \mathbf{u}_{2^a-1} \tag{6.4}$$

be the above mentioned open snakes and the separating words. It follows from the properties of the Gray code $G(k+a)$, especially from Eq. 2.9 with n replaced by $k+a$, that

$$\mathcal{S}^i = \mathcal{S}^{i-1} + \mathbf{s}_{s_i}, \quad 1 \leq i \leq 2^a, \tag{6.5}$$

where the indices s_i are the elements of the complete transition sequence of the standard Gray code $G(a)$

$$\overline{S}_a = s_1, s_2, s_3, \ldots, s_{2^a}, \tag{6.6}$$

the concrete form of which is

$$\overline{S}_a = 1, 2, 1, \ldots, a, 1, 2, 1, \ldots, a. \tag{6.7}$$

In (6.5), we identify the open snakes $\mathcal{S}^{2a}$ and $\mathcal{S}^0$. A similar rule

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \mathbf{s}_{s_{i+1}}, \quad 1 \leq i \leq 2^a \tag{6.8}$$

holds for the separating words $\mathbf{u}_i$ and $\mathbf{u}_{i-1}$in (6.4). Furthermore, the Hamming distance between $\mathbf{u}_i$ and the first word of $\mathcal{S}^{i+1}$ is equal to 1 for all relevant values of i, since these words are successive words in the Gray code generated by $\overline{S}_{k+a}(\mathcal{B}^{ext})$, according to Theorem 6.2. The Hamming distance between $\mathbf{u}_i$ and the last word of $\mathcal{S}^{i+1}$ is equal to

$$\left\| \mathbf{s}_{s_{i+1}} + \mathbf{e}_{(k+s_{i+1})_3} \right\| = \left\| \mathbf{b}_k + \mathbf{b}_{k+s_{i+1}} + \mathbf{e}_{(k+s_{i+1})_3} \right\| = \left\| \mathbf{e}_{k_3} \right\| = 1.$$

So, if we permute $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{2^a-1}$ in (6.4) in cyclic sense over one position to the right, we obtain the concatenation of 2^a closed snakes of length 2^{k+2}

$$\mathcal{S}^0, \mathbf{u}_{2^a-1}, \quad \mathcal{S}^1, \mathbf{u}_0, \quad \mathcal{S}^2, \mathbf{u}_1, \ldots, \mathcal{S}^{2^a-1}, \mathbf{u}_{2^a-2}, \tag{6.9}$$

and these snakes are identical to

$$\mathcal{S}, \quad \mathcal{S} + \mathbf{s}_1, \quad \mathcal{S} + \mathbf{s}_1 + \mathbf{s}_2, \ldots, \mathcal{S} + \sum\nolimits_{s_i \in X_i} \mathbf{s}_{s_i}, \ldots, \mathcal{S} + \mathbf{s}_a, \tag{6.10}$$

where X_i is the multiset of the first i integers of (6.6), and where the snake (list) $\mathcal{S}$ is generated by $\overline{S}_k(\mathcal{B})$. This proves the theorem. □

Example 6.3 Consider again the block list $\mathcal{B}$ of Example 4.1, and let

$$\mathcal{B}^{ext} = (\mathcal{B}, B_{12}, B_{13}, B_{14}),$$

be its extension with the three additional blocks

$$B_{12} = (0, 7, 9, 12), \quad B_{13} = (0, 7, 10, 12), \quad B_{14} = (0, 7, 0, 12).$$

Since $\mathcal{B}$ generates a snake (Example 4.1) and since $\mathcal{B}^{ext}$ satisfies the conditions of Theorem 6.2, we are entitled to apply Theorem 6.3. Hence, after having carried out the permutation $(\mathbf{u}_0\ \mathbf{u}_1\mathbf{u}_2 \cdots \mathbf{u}_7)$ of the words which are at the positions $2^{13}-1, 2\cdot 2^{13}-1, 3\cdot 2^{13}-1, \ldots, 7\cdot 2^{13}-1$, respectively, we obtain a concatenation of eight (closed) snakes which together constitute a cover of Q_{16}. This is precisely the 2^a-cover mentioned in Theorem 6.3 with $a = 3$.

We emphasize that extending $\mathcal{B}$ with the blocks

$$B_{12} = (0, 7, 9, 12), \quad B'_{13} = (0, 7, 8, 12), \quad B_{14} = (0, 7, 0, 12)$$

will not provide us with a cover. This is because the extended block list no longer satisfies the conditions of Theorem 6.2. In particular, block B_6 does not contain an integer that does not occur in blocks B_l, $l > 6$.

If some block list $\mathcal{B}^{ext}$ satisfies the conditions of Theorem 6.3, we shall say that $\mathcal{B}^{ext}$ *generates* a symmetric 2^a-cover of Q_n.

Theorem 6.4 *Let $\mathcal{B} = (B_1, B_2, \ldots, B_k)$ be a block list that satisfies the conditions of Theorem 3.2 and let k_0 be the number of blocks B_j which only satisfy condition (ii) of that theorem and not condition (i). If*

$$k_0 \le k + |I_3| + 2 - n,$$

then $\mathcal{B}$ (or an equivalent list) can be extended to a list $\mathcal{B}^{ext}$ which generates a 2^a-cover of Q_n with $a = n - k - 2$.

Proof From Theorem 3.2 we know that $\mathcal{B}$ (or an equivalent list) generates a snake in Q_n. From Theorems 6.2 and 6.3 it follows that there exists an extended list $\mathcal{B}^{ext}$ generating a 2^a-cover, if one can find $a - 1 = n - k - 3$ integers $l_3 \in I_3$ that are not in one of the k_0 blocks as described in the theorem. Since B_k is not one of these k_0 blocks, we must have

$$|I_3| - 1 - k_0 \quad \ge n - k - 3.$$

□

Example 6.4 The list $\mathcal{B} = (B_1)$ of Example 6.2 is a trivial example of Theorem 6.4. In this case, we have $n = 5$, $k = 1$, $k_0 = 0$ and $|I_3| = 2$, since $I_3 = \{2, 3\}$.

The list $\mathcal{B}$ of Examples 4.1 and 6.1 also satisfies the requirement of Theorem 6.4. Here, we have $n = 16$, $k = 11$ and $|I_3| = 4$. Since B_7 is the only block in the list that satisfies condition (ii) of Theorem 3.2 and not condition (i), it follows that $k_0 = 1$. Actually, this is a special case of a general theorem dealing with the canonical basis of the Reed-Muller code which we introduced at the beginning of this section.

Theorem 6.5 *Let $\mathcal{B}(m)$ be the canonical block list of the Reed-Muller code $R(m-2, m)$ of word length $n(= 2^m)$ and of dimension $k(= 2^m - m - 1)$. Then $\mathcal{B}(m)$ can be extended to a list $\mathcal{B}^{ext}(m)$ which generates a symmetric 2^{m-1}- cover of Q_n, for any $m \geq 3$.*

Proof We know already from Theorem 5.1 that $\mathcal{B}(m)$ generates a symmetric snake in Q_n of length 2^{k+2}, for $m \geq 3$. Moreover, if we take the I_3-integers $i_3^1, i_3^2, \ldots, i_3^{m-2}$, which are in the last blocks of $\mathcal{B}_3^1, \mathcal{B}_3^2, \ldots, \mathcal{B}_3^{m-2}$, we can extend $\mathcal{B}$ with blocks $(0, 2^{m-1} - 1, l_3, 2^m - 2^{m-2})$, with $l_3 \in\in \{i_3^1, i_3^2, \ldots, i_3^{m-2}, 0\}$. The total number of these blocks is $m-1$. This is precisely the number necessary to obtain a 2^{m-1}-cover of Q_n, according to Theorem 6.4, since $k_0 = 2^{m-2} - m + 1$. □

Corollary 6.6 *For any n which satisfies $2^{m-1} < n \leq 2^m$, one can construct a symmetric 2^{m-1}-cover of Q_n, for $m \geq 3$.*

Proof The construction can be accomplished by starting with constructing a cover of Q_{n_0} for $n_0 = 2^m$, $m \geq 3$, and next puncturing y times $1 \leq y \leq 2^{m-1} - 1$ to the pivots in blocks $B_1, B_2, \ldots, B_y$ and omitting these blocks (cf. Corollary 5.2 and the lines prior to that corollary). □

Example 6.5 To obtain a 4-cover of Q_5, we first take the canonical block list $\mathcal{B}(3)$ of $R(1, 3)$ (cf. Sect. 5) consisting of the blocks $B_1 = (\underline{1}, 2, 4, 7)$, $B_2 = (0, 3, 4, \underline{7})$, $B_3 = (0, \underline{2}, 4, 6)$ and $B_4 = (0, 3, 5, 6)$. Next we puncture successively to coordinate 1, 7 and 2, and omit blocks B_1, B_2 and B_3, yielding block list $\mathcal{B} = ((0, 3, 5, 6))$. Finally, applying (6.3), we obtain $\mathcal{B}^{ext} = ((0, 3, 5, 6), (0, 3, 4, 6), (0, 3, 0, 6))$. According to Corollary 5.2 and Theorem 6.3, this list generates a 4-cover of Q_5. If one relabels the coordinates 0, 3, 4, 5, 6 by 0, 1, 3, 2, 4, one obtains precisely the 4-cover of Q_5 discussed in Example 6.2.

We would like to point out that for $4 < n \leq 8$ and for $8 < n \leq 16$, Corollary 6.6 gives better results than [18], i.e. the number of snakes in the cover of Q_n is equal to 4 and to 8, respectively, whereas in [18] it is only stated that this number is upperbounded by 16. The 4-covers of Q_n for $4 < n \leq 8$ are minimal covers. This follows immediately from the values of $s(n)$, i.e. of the maximal snake length, for these n-values (cf. Sect. 1).

Even in the range $16 < n \leq 32$, one could say that Corollary 6.6 provides us with slightly better results, since the 16 covering snakes are symmetric, and so we have a *symmetric* 16-cover. Moreover, in the range $4 < n \leq 16$, it gives a supplement to a result in [2] which states that for any even integer $r \geq 4, n \geq 2$, the graph K_r^n, being the nth power of the complete graph K_r, can be covered with r^3 snakes, be it that these snakes are not all mutually disjoint.

7 Invariance group of a cover

In this final section, we define the notion of *invariance group* of a cover of Q_n by snakes. Let

$$\mathcal{C}^{(n)} := \{\mathcal{S}_1^{(n)}, \quad \mathcal{S}_2^{(n)}, \ldots, \mathcal{S}_N^{(n)}\} \tag{7.1}$$

be a family of mutually disjoint snakes of the same length covering the whole vertex set VQ_n of the hypercube Q_n. We consider permutations on VQ_n, i.e. elements of the symmetric group S_{2^n} acting on the set of the 2^n vertices of Q_n. Let $Aut(Q_n)$ be the subgroup of permutations π with the property that for every pair of vertices $\mathbf{v}, \mathbf{w} \in VQ_n$, $\{\mathbf{v}, \mathbf{w}\} \in EQ_n$, if and only if $\{\pi(\mathbf{v}), \pi(\mathbf{w})\} \in EQ_n$.

One can immediately verify that the translations over the vectors of $GF(2)^n$ induce a group of automorphisms of Q_n of order 2^n. The full automorphism group of Q_n is of size $2^n \times n!$ and is generated by the 2^n translation vectors acting on the vertices $\mathbf{v} = (v_1, v_2, \dots, v_n) \in VQ_n$ and the $n!$ permutations of the coordinates of $\mathbf{v}$. More precisely, $Aut(Q_n)$ is the wreath product of the symmetric groups S_n and S_2 (cf. [6, p. 177]).

It follows easily from the nearness condition (1.2) and the separability condition (1.3) (see Sect. 1), that a snake $\mathcal{S}$ is transformed into a snake $\pi(\mathcal{S})$, for any $\pi \in Aut(Q_n)$. Consequently, any $\pi \in Aut(Q_n)$ transforms a cover $\mathcal{C}^{(n)}$ to another cover $\pi(\mathcal{C}^{(n)})$

Next, we ask the question if there are $\pi \in Aut(Q_n)$ which transform $\mathcal{S}$ into $\mathcal{S}$ itself, or more generally, which $\pi \in Aut(Q_n)$ transform each snake $\mathcal{S}_i^{(n)}$ of the cover $\mathcal{C}^{(n)}$ into some snake $\mathcal{S}_j^{(n)}$ (the same snake or a different one) of $\mathcal{C}^{(n)}$. Of course, such elements π constitute a subgroup of $Aut(Q_n)$, which we shall call $Aut(\mathcal{C}^{(n)})$. So, we have the following definition.

Definition 7.1 The invariance group $Aut(\mathcal{C}^{(n)})$ of a cover $\mathcal{C}^{(n)}$ of Q_n is the group of those permutations of $Aut(Q_n)$ which induce a permutation of the snakes in $\mathcal{C}^{(n)}$.

Any subgroup of $Aut(\mathcal{C}^{(n)})$ will be called *an* invariance group of $\mathcal{C}^{(n)}$.

Now consider a 2^a-cover of Q_n as mentioned in Theorem 6.3. It will be clear that this cover is invariant under the translation group

$$\mathsf{G} =< \mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_a > \tag{7.2}$$

of order 2^a (cf. (6.10)). In particular, it follows that the covers of Q_n with $n = 2^m$ which can be constructed by applying Theorem 6.5, have an invariance translation group of order 2^{m-1}, generated by the $a = m - 1$ independent vectors $\mathbf{s}_1 = \mathbf{b}_k + \mathbf{b}_{k+1}$, $\mathbf{s}_2 = \mathbf{b}_k + \mathbf{b}_{k+2}, \dots, \mathbf{s}_a = \mathbf{b}_k + \mathbf{b}_{k+a}$, or equivalently by the $m - 1$ vectors $\mathbf{e}_0 + \mathbf{e}_{l_3}$, where $l_3 \in \{k_3, (k+1)_3, (k+2)_3, \dots, (k+m-2)_3\} = \{k_3, i_3^1, i_3^2, , \dots, i_3^{m-2}\}$ (cf. the proof of Theorem 6.5).

The same holds for any n with $2^{m-1} < n \leq 2^m$, when one interprets the translation vectors $\mathbf{s}_i$, $1 \leq i \leq m - 1$, as vectors in $GF(2)^n$ (after puncturing to relevant coordinates). We have to remark here that there will be no puncturing with respect to any of the coordinates from the set $\{k_3, i_3^1, i_3^2, , \dots, i_3^{m-2}\}$, since these integers did not serve as a pivot in any of the blocks of $\mathcal{B}(m)$.

Invariance translations other than the above ones will imply the existence of elements $\pi \in Aut(Q_n), \pi \neq 1$, such that $\pi(\mathcal{S}) = \mathcal{S}$. From the symmetry of the standard Gray code (cf. (1.5) with $K = 2^{n-1}$) and from the transition sequence (3.2), it follows that a translation over the vector $\mathbf{s}_0 := \mathbf{b}_{k-1} + \mathbf{b}_k$ satisfies this condition. So, we may conclude with the following result.

Theorem 7.1 *The 2^{m-1}-cover of Q_n, $2^{m1} < n \leq 2^m$, as mentioned in Corollary 6.6 has an invariance translation group of order 2^m generated by the vectors $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{m-1}$.*

References

1. Abbott H.L., Katchalski M.: On the construction of snake-in-the-box codes. Utilitas Math. **40**, 97–116 (1991).
2. Alsardary S.Y.: Further results on vertex covering of powers of complete graphs. Acta Math. Univ. Comenianae **66**(2), 261–283 (1997).
3. Casella D.A.: New lower bounds for the snake-in-the-box and the coil-in-the-box problems: using evolutionary techniques to hunt for snakes and coils. MSAI thesis (2005).
4. Casella D.A., Potter W.D.: New lower bounds for the snake-in-the-box problem: using evolutionary techniques to hunt for snakes. In: Proceedings of the 18th International Florida Artificial Intelligence Research Conference, pp. 264–269 (2005).
5. Emelyanov P.G., Lukito A.: On the maximal length of a snake in a hypercube of small dimension. Discerete Math. **218**, 51–59 (2000).
6. Harary F.: Graph Theory, 3rd edn. Addison Wesley (1972).
7. Kautz W.H.: Unit distance error checking codes. IEEE Trans. Electron. Comput. **7**, 179–180 (1958).
8. Klee V.: A method for constructing circuit codes. J. Assoc. Comput. Mach. **14**, 520–528 (1967).
9. Knuth D.E.: The art of computer programming, vol. 4 Fascicle 2. Generating all tuples and permutations. Addison Wesley (2005).
10. Kochut K.J.: Snake-in-the-box codes for dimension 7. J. Combin. Math. Combin. Comput. **20**, 175–185 (1996).
11. Lukito A.: Bounds for the length of certain types of distance preserving codes. Ph.D. thesis, Delft University of Technology (2000).
12. Lukito A., van Zanten A.J.: A new non-asymptotic upper bound for snake-in-the-box codes. J. Combin. Math. Combin. Comput. **39**, 147–156 (2001).
13. MacWilliams F.J., Sloane N.J.A.: The Theory of Error Correcting Codes. North Holland Mathematical Library (1977).
14. Paterson K.G., Tuliani G.: Some new circuit codes. IEEE Trans. Inform. Theory **44**, 1305–1309 (1998).
15. Potter W.D., et al.: Using the genetic algorithm to find snake-in-the-box codes. In: Proceedings of the 7th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. United States Austin, Texas, pp. 421–426 (1994).
16. Snevily H.S.: The snake-in-the-box problem, a new upper bound. Discrete Math. **133**, 307–304 (1994).
17. Solov'jeva F.I.: An upper bound for the length of a cycle in an n-dimensional unit cube, Diskret. Analiz. **45**, 71–76 (1987).
18. Wojciechowski J.: Covering the hypercube with a bounded number of disjoint snakes. Combinatorica **14**, 491–496 (1994).
19. van Zanten A.J.: Minimal-change order and separability in linear codes. IEEE Trans. Inform. Theory **39**, 1988–1989 (1993).
20. van Zanten A.J., Lukito A.: Construction of certain cyclic distance-preserving codes having linear-algebraic characteristic. Des. Codes Cryptogr. **16**, 185–199 (1999).
21. van Zanten A.J., Haryanto L.: Covers and Near-Covers of the Hypercube Q_{16} by Symmetric Snakes, Report CS 06-01. Department of Computer Science, Universiteit Maastricht (2006).
22. Zémor G.: An upper bound of the size of snakes. Combinatorica **17**, 287–298 (1997).