Efficient Non-Recursive Design of Second-Order Spectral-Null Codes

Luca G. Tallini, Danilo Pelusi, Raffaele Mascella, Laura Pezza, Samir Elmougy, and Bella Bose, *Fellow, IEEE*

***Abstract*—A new efficient design of second-order spectral-null (2-OSN) codes is presented. The new codes are obtained by applying the technique used to design parallel decoding balanced (i.e., 1-OSN) codes to the random walk method introduced by some of the authors for designing 2-OSN codes. This gives new non-recursive efficient code designs, which are less redundant than the code designs found in the literature. In particular, if $k \in \mathbb{N}$ is the length of a 1-OSN code then the new 2-OSN coding scheme has length $n = k + r \in \mathbb{N}$ with an extra redundancy of $r \simeq 2\log_2 k + (1/2)\log_2\log_2 k - 0.174$ check bits, with k and r even and n multiple of 4. The whole coding process requires $O(k \log k)$ bit operations and $O(k)$ bit memory elements.**

***Index Terms*—High order spectral null codes, balanced codes, Knuth's complementation method, parallel decoding scheme, optical and magnetic recording.**

I. Introduction

Let $\Phi = \{-1, +1\}$ be the bipolar alphabet. The set of qth-order spectral-null words $\mathcal{SN}(n, q) \subseteq \Phi^n$ is defined as [5], [6], [14],

$$\mathcal{SN}'(n,q) \overset{\text{def}}{=} \left\{ X \in \Phi^n \,\middle|\, \begin{array}{l} m_i(X) = 0, \\ \text{for all integer } i \in [0, q-1] \end{array} \right\}; \quad (1)$$

where $m_i(X) \overset{\text{def}}{=} x_1 1^i + x_2 2^i + \ldots + x_n n^i = \sum_{j=1}^{n} x_j j^i$ is the ith moment (or, the m_i-weight) of the word $X = x_1 x_2 \ldots x_n \in \Phi^n$, and sums and products are over the real numbers. Any word in $\mathcal{SN}(n, q)$ is called qth-order spectral-null word (briefly, a q-OSN word). In the following, if $m_i(X) = 0$ then we will say that the word X is m_i-balanced. A binary code $\mathcal{C}$ is a qth-order spectral-null code with $\tilde{k}$ information bits and length n (briefly, a q-OSN$(n, \tilde{k})$ code) if, and only if

1) $\mathcal{C}$ is a subset of $\mathcal{SN}(n, q)$, and
2) $\mathcal{C}$ has $2^{\tilde{k}}$ codewords.

When $q = 1$, these codes coincide with the so-called balanced or DC-free block codes [1]–[3], [5], [7], [10]–[14], [17]–[19], [21], [23]–[25], [28]. For values of q greater than 1, the q-OSN$(n, \tilde{k})$ codes are considered in digital recording to achieve a better rejection of the low frequency components of the transmitted signal and enhancing the error correction capability of codes used in partial-response channels [6], [14]. The q-OSN codes can be also considered over the binary alphabet $\mathbb{Z}_2 = \{0, 1\}$ [20]. In fact, by replacing the symbol -1 with 0 and $+1$ with 1, the set $\mathcal{SN}(n, q)$ becomes equivalent to the set $\mathcal{SN}'(n, q) \subseteq \mathbb{Z}_2^n$

$$\mathcal{SN}'(n,q) \overset{\text{def}}{=} \left\{ X \in \mathbb{Z}_2^n \,\middle|\, \begin{array}{c} m_i(X) = \sum_{j=1}^{n} x_j j^i = \frac{1}{2}\sum_{j=1}^{n} j^i, \\ \text{for all integer } i \in [0, q-1] \end{array} \right\};$$

where the sums and products are done in the real field $\mathbb{R}$. Since $\mathcal{SN}(n, q)$ and $\mathcal{SN}'(n, q)$ are equivalent, in the rest of the paper $\mathcal{SN}(n, q)$ is used for $\mathcal{SN}'(n, q)$. For example,

$$\mathcal{SN}(n,2) = \left\{ X \in \mathbb{Z}_2^n \,\middle|\, \begin{array}{l} m_0(X) = \sum_{j=1}^{n} x_j = \frac{n}{2} \quad \text{and} \\ m_1(X) = \sum_{j=1}^{n} x_j j = \frac{n(n+1)}{4} \end{array} \right\}$$

and $\mathcal{SN}(n, 2) \neq \emptyset$ if, and only if, n is multiple of 4 [20].

The code design problem is to convert the information words into qth-order spectral-null words using the minimum possible redundancy. For $q = 2$, this minimum redundancy is [20]

$$r_{min}(k) = 2\log_2 n - 1.141.$$

Also, the conversion should be done so that the encoding and decoding processes are as computationally simple as possible. Efficient design of qth-order spectral-null codes for $q = 2$, has been considered in [8], [9], [14], [20], [22], [26], [27] and, for $q = 3$, in [15].

In this paper we are concerned with designing 2-OSN codes whose encoding and decoding functions can be computed combining the Knuth's optimal refined parallel decoding scheme for balanced codes given in [1], [11], [12], and [19]

Manuscript received July 4, 2015; revised March 29, 2016; accepted March 30, 2016. Date of publication April 20, 2016; date of current version May 18, 2016. This work was supported in part by the National Science Foundation within the Division of Computing and Communication Foundations under Grant CCF-1117215 and Grant CCF-1423656, in part by Italian MIUR under Grant PRIN 2009-2009EL7424, in part by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award Number (10-INF1165-02). This paper was partially presented at the the 2013 IEEE International Symposium on Information Theory.

L. G. Tallini, D. Pelusi, and R. Mascella are with the Facoltà di Scienze della Comunicazione, Università degli Studi di Teramo, Teramo 64100, Italy (e-mail: ltallini@unite.it; dpelusi@unite.it; rmascellag@unite.it).

L. Pezza is with the Dipartimento di Scienze di Base e Applicate per l'Ingegneria, Università degli Studi di Roma "La Sapienza", Rome 00161, Italy (e-mail: pezza@dmmm.uniroma1.it).

S. Elmougy is with the Department of Computer Science, Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt, and also with the Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: mougy@mans.edu.eg).

B. Bose is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA (e-mail: bose@eecs.orst.edu).

Communicated by M. Schwartz, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2016.2555322

with the random walk method for 2-OSN codes given in [9], [20], and [26]. Here we assume that the information words are already m_0-balanced words of even length k (i.e., the information words belongs to $\mathcal{SN}(k,1) \neq \emptyset$) by using a 1-OSN$(k,\tilde{k})$ code design. Given $k, \mu_0, \mu_1 \in \mathbb{N}$, let

$$S(k,\mu_0) \overset{\text{def}}{=} \{X \in \mathbb{Z}_2^k : m_0(X) = \mu_0\} \subseteq \mathbb{Z}_2^k$$

and

$$S(k,\mu_0,\mu_1) \overset{\text{def}}{=} \{X \in \mathbb{Z}_2^k\colon m_0(X) = \mu_0 \text{ and } m_1(X) = \mu_1\} \subseteq S(k,\mu_0).$$

Now, given an m_0-balanced word $X = x_1x_2\ldots x_k \in S(k,k/2)$ (i.e., such that $m_0(X) = k/2 \in \mathbb{N}$), the idea is to "walk randomly" towards the reverse, $X^R \overset{\text{def}}{=} x_kx_{k-1}\ldots x_1$, of X, by exchanging adjacent bits, until a word $X^{(d_{h_{bal}})}$ is reached, with $d_h, h, h_{bal} \in \mathbb{N}$, where this word satisfies a specific property for its second moment. Note that permutations of adjacent bits in the word X do not alter the value of its first moment, while its second moment may change with a variation of $+1$, -1, or 0.

Moreover, a family of sets of check words, $\mathcal{CS}$, is built so that each set of check words, $\Gamma_h \in \mathcal{CS}$, identifies a specific step d_h of the random walk, where $h = 0, 1, \ldots, |\mathcal{CS}| - 1$. As such, each set Γ_h of the family of check words defines an encoding function $\langle\, \Gamma_h \rangle$. And, as the main theorem states (Theorem 2), there always exists at least one step $d_{h_{bal}}$ such that $X^{(d_{h_{bal}})} = \langle\, \Gamma_{h_{bal}} \rangle(X)$ concatenated with one of the check words identifying $\langle\, \Gamma_h \rangle$, becomes a 2-OSN word. As in [19] for 1-OSN codes, such possibly many indices h_{bal}, for a given code design, are referred to as the 2-OSN balancing indices of X. Here we let $h_{bal}(X) \subseteq \mathbb{N}$ indicate the set of all possible balancing indices of X. Now, a check word $C = C(X) \in \mathbb{Z}_2^r$, $r \in \mathbb{N}$, is appended to obtain a $n = k + r$ bit codeword

$$\mathcal{E}_2(X) = X^{(d_{h_{bal}})}C(X) \in \mathbb{Z}_2^n$$

as encoding of X. In order to implement this strategy, such check C must be chosen so that the properties 1), 2) and 3) below hold.

1) The codeword $\mathcal{E}_2(X)$ is m_0-balanced; that is,
$$m_0(\mathcal{E}_2(X)) = m_0\left(X^{(d_{h_{bal}})}\right) + m_0(C) = \frac{n}{2} \in \mathbb{N}.$$
Note that, since n is a multiple of 4 (otherwise, we have $\mathcal{SN}(n,2) = \emptyset$), if k is even and $X \in S(k,k/2)$ then r is even and $C \in S(r,r/2)$.
2) The codeword $\mathcal{E}_2(X)$ is m_1-balanced; that is,
$$m_1\left(\mathcal{E}_2(X)\right) = m_1\left(X^{(d_{h_{bal}})}C\right) = \frac{n(n+1)}{4} \in \mathbb{N}.$$
Thus, 1) and 2) make $\mathcal{E}_2(X)$ a 2nd-order spectral-null word.
3) The original information word X can be recovered from $X^{(d_{h_{bal}})}$ and C.

In Section II, to fulfill the above properties, we start with the 1-OSN words of even length, $k \in 2\mathbb{N}$ and then we encode these words into 2-OSN words whose length is $n \in 4\mathbb{N}$. This last step is obtained by adapting the coding scheme given in [1] and [19] for balanced codes to fit the random walk coding scheme given in [20] for 2-OSN codes. The combination of the two methods gives efficient non-recursive code designs for any value of the parameters $k \in 2\mathbb{N}$, $r = n - k \in 2\mathbb{N}$ and $n \in 4\mathbb{N}$, provided that

$$\frac{k(k-1)}{2} \le \binom{r}{r/2} \iff k \le \frac{1+\sqrt{1+8\binom{r}{r/2}}}{2}. \qquad (2)$$

Note that, if

$$k \simeq \frac{1}{2}\left(1+\sqrt{1+8\binom{r}{r/2}}\right)$$

then

$$r \simeq 2\log_2 k + (1/2)\log_2\log_2 k - 0.174 \qquad (3)$$

because of Stirling's approximation. As shown in Table II, we get 2-OSN codes which are considerably less redundant than the codes found in the literature. To our knowledge, all code designs found in the literature are recursive and use only 2-OSN words as check words. Here, on the other hand, the code designs are non-recursive and make a potential good use of all possible 1-OSN words as check words. Hence, the information rate improvement has two reasons. First, the non-recursiveness of the code designs makes sure that any unuseful redundancy does not add up at every encoding recursion step. Second, the presented code designs make available many more check words to use in the design. In any case, the code rate improvement is of the order of $2\log_2\log_2 k + O(\log\log\log k)$ if k is choosen to be the length of an optimal 1-OSN/balanced code. This can be seen by comparing (3) with the formula for the code length (6) in [20]. In Section III, we show that the whole coding process can be implemented with $O(k\log k)$ bits operations and using $O(k)$ memory bits. In Section IV, some concluding remarks are given.

II. The Proposed Refined Coding Scheme

The main idea of the code design is to convert a balanced data word into an "almost 2-OSN" word, using an appropriate function from a set of "m_1-balancing functions", and append a check word. This check word 1) "encodes" which encoding function is used in the encoding process $\mathcal{E}_2$, and 2) corrects any further m_1-imbalance of the almost 2-OSN word. To decode a codeword, the receiver simply applies, to the m_0-balanced information part, the inverse of the function encoded by the check part.

The coding scheme is captured by the following concept of the set of m_1-balancing functions.

Definition 1 (Set of m_1-Balancing Functions): Let k, $r \in 2\mathbb{N}$ be given so that $n \overset{\text{def}}{=} k + r \in 4\mathbb{N}$. Also, let

$$\mathcal{CS} \overset{\text{def}}{=} \{\Gamma_0, \Gamma_1, \ldots, \Gamma_{p-1}\},$$

be a family of $p \in \mathbb{N}$ non-empty subsets of the set of all the r-bit m_0-balanced check words $S(r,r/2)$. Also, for all indices $h \in [0, p-1]$, let

$$\langle\, \Gamma_h \rangle\colon S(k,k/2) \to S(k,k/2)$$

indicate a function from the set of all k-bit m_0-balanced data words, $S(k, k/2)$, into itself. We say that the set of functions

$$\mathcal{B} \overset{\text{def}}{=} \{\langle \Gamma_h \rangle : \quad h \in [0, p-1]\}$$

is a set of m_1-balancing functions if, and only if, the following conditions hold.

1) *The sets Γ_h are pair-wise disjoint; i.e.,*

$$\Gamma_i \cap \Gamma_j = \emptyset \iff i \neq j.$$

Thus, from $C \in \Gamma_h$ it is possible to unambiguously recover $h \in [0, p-1]$.

2) *For every m_0-balanced information word $X \in S(k, k/2)$ there exists one $h_{bal} \in [0, p-1]$ and $C_{h_{bal}} \in \Gamma_{h_{bal}}$ such that*

$$m_1\left(\langle \Gamma_{h_{bal}} \rangle(X)\ C_{h_{bal}}\right) = \frac{n(n+1)}{4}.$$

In the sequel, we refer to an index h_{bal} as an "m_1-balancing index" of X and we let $h_{bal}(X) \subseteq [0, p-1]$ indicate the set of all possible balancing indices of X. Note that, for all $X \in S(k, k/2)$,

$$\begin{aligned} m_0\left(\langle \Gamma_{h_{bal}} \rangle(X)\ C_{h_{bal}}\right) &= m_0\left(\langle \Gamma_{h_{bal}} \rangle(X)\right) + m_0\left(C_{h_{bal}}\right) \\ &= \frac{k}{2} + \frac{r}{2} = \frac{n}{2}. \end{aligned}$$

3) *For all indices $h \in [0, p-1]$, the function $\langle \Gamma_h \rangle$ is one-to-one so that, from h and $Y = \langle \Gamma_h \rangle(X)$ it is possible to unambiguously recover X.*

Given a set of m_1-balancing functions, every m_0-balanced data word $X \in S(k, k/2)$ is encoded as

$$\mathcal{E}_2(X) = \langle \Gamma_{h_{bal}} \rangle(X)\ C_{h_{bal}}$$

where $h_{bal} \in h_{bal}(X)$ is an "m_1-balancing index" of X; for example, the smallest index in the set $h_{bal}(X)$. We readily note that, to give a well defined set of m_1-balancing functions $\mathcal{B}$, it is required to define

a) the partition $\mathcal{CS}$, and

b) each injective function $\langle \Gamma_h \rangle : S(k, k/2) \to S(k, k/2)$ in $\mathcal{B}$;

and these two requirements are dependent on one another by the condition 2) of Definition 1. Now, a possible construction for such balancing functions $\langle \Gamma \rangle \in \mathcal{B}$, with $\Gamma \in \mathcal{CS}$, is explained below.

First, given

$$X = x_1 x_2 x_3 \ldots x_i \ldots x_j \ldots x_{k-1} x_k$$

let

$$X^{(i,j)} = x_1 x_2 x_3 \ldots \underline{x_j} \ldots \underline{x_i} \ldots x_{k-1} x_k$$

be the word obtained form X by exchanging the ith bit with the jth bit, and

$$X^R = \underline{x_k x_{k-1} \ldots x_j \ldots x_i \ldots x_3 x_2 x_1}$$

be the reverse of X. The code design is based on the following observations [20].

Observation 1: The m_0-weight, $m_0(X)$, does not change if the bits of X are permuted.

Observation 2: The relation

$$m_1(X) + m_1(X^R) = (k+1) m_0(X)$$

holds. So, if $m_0(X) = k/2 \in \mathbb{N}$ then

$$m_1(X) \leq \frac{k(k+1)}{4} \iff m_1(X^R) \geq \frac{k(k+1)}{4}.$$

Observation 3: The following relation holds

$$m_1\left(X^{(i,i+1)}\right) = m_1(X) + (x_i - x_{i+1}).$$

This implies that by exchanging two consecutive bits of X, the value of m_1 either decreases by 1 or remains the same or increases by 1.

So, as in [20], let us consider the sequence of $k(k-1)/2+1$ words,

$$\begin{aligned}
X^{(0)} &\overset{\text{def}}{=} X && = X, \\
X^{(1)} &\overset{\text{def}}{=} \left(X^{(0)}\right)^{(1,2)}, \\
X^{(2)} &\overset{\text{def}}{=} \left(X^{(1)}\right)^{(2,3)}, \\
&\vdots \\
X^{(k-1)} &\overset{\text{def}}{=} \left(X^{(k-2)}\right)^{(k-1,k)}, \\
X^{(k)} &\overset{\text{def}}{=} \left(X^{(k-1)}\right)^{(k-2,k-1)}, \\
X^{(k+1)} &\overset{\text{def}}{=} \left(X^{(k)}\right)^{(k-3,k-2)}, \\
&\vdots \\
X^{(2k-3)} &\overset{\text{def}}{=} \left(X^{(2k-4)}\right)^{(1,2)} && = X^{(1,k)}, \\
X^{(2k-2)} &\overset{\text{def}}{=} \left(X^{(2k-3)}\right)^{(2,3)}, \\
X^{(2k-1)} &\overset{\text{def}}{=} \left(X^{(2k-2)}\right)^{(3,4)}, \\
&\vdots \\
X^{(3k-6)} &\overset{\text{def}}{=} \left(X^{(3k-7)}\right)^{(k-2,k-1)}, \\
X^{(3k-5)} &\overset{\text{def}}{=} \left(X^{(3k-6)}\right)^{(k-3,k-2)}, \\
X^{(3k-4)} &\overset{\text{def}}{=} \left(X^{(3k-5)}\right)^{(k-4,k-3)}, \\
&\vdots \\
X^{(4k-10)} &\overset{\text{def}}{=} \left(X^{(4k-9)}\right)^{(2,3)} && = \left(X^{(1,k)}\right)^{(2,k-1)}, \\
&\vdots \\
X^{(k(k-1)/2)} &\overset{\text{def}}{=} \left(X^{(k(k-1)/2)-1}\right)^{(k/2,k/2+1)} && = X^R.
\end{aligned}$$

Example 1 (Word Sequence for $k = 4$): For example, if $k = 4$ and $X = x_1 x_2 x_3 x_4$ then the sequnece has cardinality

$k(k-1)/2+1=7$ *and*

$$\begin{aligned}
X^{(0)} &= x_1x_2x_3x_4 &&= X,\\
X^{(1)} &= \underline{x_2x_1}x_3x_4, &&\\
X^{(2)} &= x_2\underline{x_3x_1}x_4, &&\\
X^{(3)} &= x_2x_3\underline{x_4x_1}, &&\\
X^{(4)} &= x_2\underline{x_4x_3}x_1, &&\\
X^{(5)} &= \underline{x_4x_2}x_3x_1 &&= X^{(1,4)},\\
X^{(6)} &= \underline{x_4x_3x_2x_1} &&= \left(X^{(1,4)}\right)^{(2,3)} = X^R.
\end{aligned}$$

The above sequence $\{X^{(i)}\}_{i=0,\dots,k(k-1)/2}$ is defined by a particular sequence of exchanging of consecutive bits of X, with $X^{(0)}=X$ and $X^{(k(k-1)/2)}=X^R$. Since

$$m_1\left(X^{(i+1)}\right)-m_1\left(X^{(i)}\right)\in\{-1,0,+1\}$$

for all integer $i\in[0,k(k-1)/2]$, it follows that the sequence

$$\left\{\left(i,m_1\left(X^{(i)}\right)\right):i=0,1,\dots,\frac{k(k-1)}{2}\right\}$$

represents a "random walk" from the point $(0,m_1(X))$ to the point $\left(k(k-1)/2,m_1\left(X^R\right)\right)$ and satisfies the following key properties.

For all $X\in S(k,k/2)$,

there exists an integer

$$i_{sb}\in[0,k(k-1)/2+[(k/2)\bmod 2])$$

such that

$$m_1\left(X^{(i_{sb})}\right)=\left\lfloor\frac{k(k+1)}{4}\right\rfloor \qquad \left(\text{or } m_1\left(X^{(i_{sb})}\right)=\left\lceil\frac{k(k+1)}{4}\right\rceil\right). \tag{4}$$

In the following, we may refer to such an integer i_{sb} as a "simple m_0-balancing index" for the given m_0-balanced word X;

and

For all $X\in S(k,k/2)$ and

for all integers $i_1,i_2\in[0,k(k-1)/2]$,

$$m_1\left(X^{(i_2)}\right)\in\left[m_1\left(X^{(i_1)}\right)-|i_2-i_1|,\; m_1\left(X^{(i_1)}\right)+|i_2-i_1|\right]. \tag{5}$$

In [20], a simple coding scheme for 2-OSN codes is given by defining the balancing functions as $X\to X^{(i)}$, where $i\in[0,k(k-1)/2)$. In particular, given an m_0-balanced data word $X\in S(k,k/2)$, with $k\in 4\mathbb{N}$, the codeword associated with X is recursively defined as

$$\mathcal{E}(X)=X^{(i_{sb})}\,\mathcal{E}(\text{binary representation of } i_{sb} \text{ given in (4)}).$$

Now we come to the definition of the set $\mathcal{CS}$ satisfying 1) of Definition 1. Let $p\in\mathbb{N}$ and consider the partition $\{\Gamma_h\}_{h=0,1,\dots,p-1}$ of a subset of $S(r,r/2)$ into non-empty sets defined as follows. The set Γ_0 contains exactly one word of the set $S(r,r/2,\mu_1)$ for each possible value μ_1 in the "m_1-image" of $S(r,r/2)$ defined to be

$$m_1\left(S(r,r/2)\right)\overset{\text{def}}{=}\{m_1(C):\,C\in S(r,r/2)\}.$$

Then let Γ_1 contain exactly one word of the set $S(r,r/2,\mu_1)-\Gamma_0$ for each possible value $\mu_1\in m_1(S(r,r/2)-\Gamma_0)$. In general, the check word partition $\mathcal{CS}$ is defined by the following simple constructive rule

For $h=0,1,2,\dots,p-1$,

for all integer

$$\mu_1\in m_1\left(S(r,r/2)-\bigcup_{j=0}^{h-1}\Gamma_j\right), \tag{6}$$

the set Γ_h contains exactly one check word in $S(r,r/2,\mu_1)-\bigcup_{j=0}^{h-1}\Gamma_j$.

Note that such definition is well defined and implies

$$p\le\max_{\mu}|S(r,r/2,\mu)|.$$

Example 2 (Case $k=6$, $r=6$ and $p=3$): For example, if $r=6$ and $p=\max_\mu|S(6,3,\mu)|=3$ then

$$\begin{aligned}
S(r=6,r/2=3,\mu_1=6)&=\{111000\},\\
S(r=6,r/2=3,\mu_1=7)&=\{110100\},\\
S(r=6,r/2=3,\mu_1=8)&=\{101100,110010\},\\
S(r=6,r/2=3,\mu_1=9)&=\{011100,101010,110001\},\\
S(r=6,r/2=3,\mu_1=10)&=\{011010,100110,101001\},\\
S(r=6,r/2=3,\mu_1=11)&=\{010110,011001,100101\},\\
S(r=6,r/2=3,\mu_1=12)&=\{001110,010101,100011\},\\
S(r=6,r/2=3,\mu_1=13)&=\{001101,010011\},\\
S(r=6,r/2=3,\mu_1=14)&=\{001011\},\\
S(r=6,r/2=3,\mu_1=15)&=\{000111\};
\end{aligned}$$

so that, for example, $\mathcal{CS}=\{\Gamma_0,\Gamma_1,\Gamma_2\}$ where

$$\Gamma_0=\left\{\begin{matrix}111000,\\110100,\\101100,\\011100,\\011010,\\010110,\\001110,\\001101,\\001011,\\000111\end{matrix}\right\},\quad \Gamma_1=\left\{\begin{matrix}110010,\\101010,\\100110,\\011001,\\010101,\\010011\end{matrix}\right\},$$

$$\Gamma_2=\left\{\begin{matrix}110001,\\101001,\\100101,\\100011\end{matrix}\right\}. \tag{7}$$

Here, the combinatorial problem is very similar to the coding problem for balanced codes solved with the parallel decoding techniques given in [1] and [19]. In fact, the cardinalities of the constant m_1-weight codes contained in $S(r,\mu_0)$ show a "bell shape behaviour" similar to the binomial distribution as stated in Theorem 3 in the Appendix. In our case, if $\mathcal{CS}=\{\Gamma_h\}_{h\in[0,p-1]}$ is defined by (6) then Theorem 3 for $\mu_0=r/2$ implies that for all integer $h\in[0,p-1]$, the statements in the following observations hold.

Observation 4: The m_1-image of the words in Γ_h, $m_1(\Gamma_h)$, is an integer interval, say

$$m_1(\Gamma_h) \overset{\text{def}}{=} [\alpha_h, \beta_h].$$

Observation 5: If $h > 0$ then

$$m_1(\Gamma_h) = [\alpha_h, \beta_h] \subseteq [\alpha_{h-1}, \beta_{h-1}] = m_1(\Gamma_{h-1}).$$

Observation 6: If $\alpha \overset{\text{def}}{=} \min_{C \in S(r,r/2)} m_1(C)$ and $\beta \overset{\text{def}}{=} \max_{C\in S(r,r/2)} m_1(C)$ then the interval $m_1(\Gamma_h) = [\alpha_h, \beta_h]$ is symmetric with respect to

$$\mu_{mean} = \frac{\alpha+\beta}{2} = \frac{\alpha_h+\beta_h}{2} = \frac{r(r+1)}{4} \in \mathbb{R},$$

has a width of $(\beta_h - \alpha_h) = |\Gamma_h| - 1$ where

$$|\Gamma_h| - 1 \equiv \beta_h - \alpha_h \equiv |\Gamma_0| - 1 \equiv \beta - \alpha \equiv \left(\frac{r}{2}\right)^2 \equiv \left(\frac{r}{2}\right) \bmod 2.$$

Hence,

$$\begin{aligned} m_1(\Gamma_h) &= [\alpha_h, \beta_h] \\ &= \left[\frac{\alpha_h+\beta_h}{2} - \frac{\beta_h-\alpha_h}{2}, \frac{\alpha_h+\beta_h}{2} + \frac{\beta_h-\alpha_h}{2}\right] \\ &= \left[\frac{r(r+1)}{4} - \frac{|\Gamma_h|-1}{2}, \frac{r(r+1)}{4} + \frac{|\Gamma_h|-1}{2}\right] \\ &= \left[\left\lfloor\frac{r(r+1)}{4}\right\rfloor - \left\lfloor\frac{|\Gamma_h|-1}{2}\right\rfloor,\right. \\ &\qquad \left.\left\lceil\frac{r(r+1)}{4}\right\rceil + \left\lfloor\frac{|\Gamma_h|-1}{2}\right\rfloor\right]. \end{aligned} \tag{8}$$

Observation 7: Let $r \in \mathbb{N}$ and, for all integer $\mu_0, \mu_1 \in \mathbb{N}$,

$$s(r, \mu_0, \mu_1) \overset{\text{def}}{=} |S(r, \mu_0, \mu_1)|. \tag{9}$$

The number of m_1-balancing functions is,

$$\begin{aligned} p \le \max_{\mu\in[\alpha,\beta]} |S(r, r/2, \mu)| &= s\left(r, r/2, \left\lfloor\frac{r(r+1)}{4}\right\rfloor\right) \\ &= s\left(r, r/2, \left\lceil\frac{r(r+1)}{4}\right\rceil\right). \end{aligned}$$

Finally, given the above partition

$$\mathcal{CS} = \{\Gamma_h : h \in [0, p-1]\},$$

we come to the definition of each function $\langle\,\Gamma_h\rangle \in \mathcal{B}$. This is done as in [1], [12], and [19]. Namely, define the following p natural numbers,

$$\begin{aligned} d_h &\overset{\text{def}}{=} \begin{cases} 0 & \text{if } h = 0, \\ d_{h-1} + \left\lfloor\frac{|\Gamma_{h-1}|}{2}\right\rfloor + \left\lceil\frac{|\Gamma_h|}{2}\right\rceil & \text{if } h \in [1, p-1], \end{cases} \\ &= \begin{cases} 0 & \text{if } h = 0, \\ \left\lfloor\frac{|\Gamma_0|}{2}\right\rfloor + \sum_{j=1}^{h-1} |\Gamma_j| + \left\lceil\frac{|\Gamma_h|}{2}\right\rceil & \text{if } h \in [1, p-1]; \end{cases} \end{aligned} \tag{10}$$

their associated integer intervals,

$$I_h \overset{\text{def}}{=} \left[d_h - \left\lceil\frac{|\Gamma_h|}{2}\right\rceil + 1, d_h + \left\lfloor\frac{|\Gamma_h|}{2}\right\rfloor\right],$$

for all integer $h \in [0, p-1]$; and, also let,

$$\begin{aligned} I_p &\overset{\text{def}}{=} \left[d_{p-1} + \left\lfloor\frac{|\Gamma_{p-1}|}{2}\right\rfloor + 1, \sum_{h=0}^{p-1} |\Gamma_h|\right) \\ &= \left[\left\lfloor\frac{|\Gamma_0|}{2}\right\rfloor + \sum_{h=1}^{p-1} |\Gamma_h| + 1, \sum_{h=0}^{p-1} |\Gamma_h|\right). \end{aligned} \tag{11}$$

Note that, from the construction (10),

1) each interval is centered in $d_h + [(|\Gamma_h| - 1) \bmod 2]/2 \in \mathbb{R}$, for all $h \in [0, p-1]$; and
2) the family $\{I_h : h \in [0, p]\}$ partitions the necessary range of the random walk index i, which, from (4), is given by the integer interval $[0, k(k-1)/2 + [(k/2) \bmod 2])$. This, if $k(k-1)/2 + [(k/2) \bmod 2] \le \sum_{h=0}^{p-1} |\Gamma_h|$.

Now, given the d_h, define the associated balancing functions

$$\begin{aligned} &\langle\,\Gamma_h\rangle : S(k, k/2) \to S(k, k/2) \quad \text{as} \\ &\langle\,\Gamma_h\rangle(X) \overset{\text{def}}{=} X^{(d_h)}, \quad \forall h \in [0, p-1]. \end{aligned} \tag{12}$$

Example 3 [Case $k = 6$, $r = 6$ and $p = 3$ (Continued)]: In (7), where $r = 6$ and $p = 3$, from (7), (10) and (12), we have

$$\begin{aligned} &d_0 = 0, \\ &d_1 = d_0 + |\Gamma_0|/2 + |\Gamma_1|/2 = 0 + 5 + 3 = 8 \textit{ and} \\ &d_2 = d_1 + |\Gamma_1|/2 + |\Gamma_2|/2 = 8 + 3 + 2 = 13. \end{aligned}$$

So, if $k = 6$ then

$$\begin{aligned} I_0 &\overset{\text{def}}{=} [-4, 5] = \{-4, -3, -2, -1, \mathbf{0}, 1, 2, 3, 4, 5\}, \\ I_1 &\overset{\text{def}}{=} [6, 11] = \{6, 7, \mathbf{8}, 9, 10, 11\}, \\ I_2 &\overset{\text{def}}{=} [12, 15] = \{12, \mathbf{13}, 14, 15\}, \\ I_3 &\overset{\text{def}}{=} [16, 20) = \{16, 17, 18, 19\}, \end{aligned}$$

the random walk index

$$i \in [0, 15] = [0, 5] \cup [6, 11] \cup [12, 15] \subseteq I_0 \cup I_1 \cup I_2 \cup I_3,$$

and

$$\begin{aligned} \langle\,\Gamma_0\rangle(X) &\overset{\text{def}}{=} x_1x_2x_3x_4x_5x_6 = X^{(\mathbf{0})}, \\ \langle\,\Gamma_1\rangle(X) &\overset{\text{def}}{=} x_2\underline{x_6x_3}x_4x_5\underline{x_1} = X^{(\mathbf{8})}, \\ \langle\,\Gamma_2\rangle(X) &\overset{\text{def}}{=} \underline{x_6}x_3\underline{x_5x_4}\underline{x_2x_1} = X^{(\mathbf{13})}. \end{aligned}$$

We need to show that the set $\mathcal{B}$ just defined is indeed a well defined set of m_1-balancing functions. This is done in Theorem 2 below. First, note that to obtain the codewords a check word must be concatenated to some $Y = \left\langle\,\Gamma_{h_{bal}}\right\rangle(X)$. So, the following theorem concerning the moments of the concatenation of words holds.

Theorem 1: For all $Y \in \mathbb{Z}_2^k$ and $C \in \mathbb{Z}_2^r$, the following relations hold.

$$m_0(YC) = m_0(Y) + m_0(C) \tag{13}$$

and

$$m_1(YC) = m_1(Y) + m_1(C) + km_0(C). \tag{14}$$

Proof: With regard to (14), if $Y = y_1 y_2 \dots y_k \in \mathbb{Z}_2^k$ and $C = c_1 c_2 \dots c_r \in \mathbb{Z}_2^r$ then

$$\begin{aligned} m_1(YC) &= \sum_{j=1}^{k} y_j j + \sum_{j=k+1}^{k+r} c_{j-k} j \\ &= m_1(Y) + \sum_{j=1}^{r} c_j (k+j) \\ &= m_1(Y) + k \sum_{j=1}^{r} c_j + \sum_{j=1}^{r} c_j j \\ &= m_1(Y) + k m_0(C) + m_1(C). \end{aligned}$$

■

Also, we need the following definition.

Definition 2 (m_1-Crossing Index): Let $k, r \in 2\mathbb{N}$ and $n \overset{\text{def}}{=} k + r \in 4\mathbb{N}$. An index

$$i_{cx} \in \left[0, \frac{k(k-1)}{2} + \left[\binom{k}{2} \bmod 2\right]\right)$$

is an "m_1-crossing index" for $X \in S(k, k/2)$ if, and only if, "i_{cx} defines a point where the random walk

$$RW(X) \overset{\text{def}}{=} \left\{ \left(i, m_1\left(X^{(i)}\right)\right) : i = 0, 1, \dots, \frac{k(k-1)}{2} + \left[\binom{k}{2} \bmod 2\right] - 1 \right\} \quad (15)$$

crosses the m_1 middle value of $k(k+1)/4 \in \mathbb{R}$" – more precisely, if, and only if, one of the following two conditions hold.

1) *$k, r \in 4\mathbb{N}$ and i_{cx} is a simple m_1-balancing index for X according to (4); that is,*

$$m_1\left(X^{(i_{cx})}\right) = \frac{k(k+1)}{4} \in \mathbb{N};$$

2) *$k, r \in 4\mathbb{N} + 2$ and i_{cx} is a simple decreasing m_1-balancing index for X; that is,*

$$m_1\left(X^{(i_{cx}-1)}\right) = \left\lceil \frac{k(k+1)}{4} \right\rceil > \left\lfloor \frac{k(k+1)}{4} \right\rfloor = m_1\left(X^{(i_{cx})}\right). \quad (16)$$

3) *$k, r \in 4\mathbb{N}+2$ and i_{cx} is a simple increasing m_1-balancing index for X; that is,*

$$m_1\left(X^{(i_{cx}-1)}\right) = \left\lfloor \frac{k(k+1)}{4} \right\rfloor < \left\lceil \frac{k(k+1)}{4} \right\rceil = m_1\left(X^{(i_{cx})}\right). \quad (17)$$

We note that, for all $X \in S(k, k/2)$ there always exists at least one m_1-crossing index, i_{cx}, of X because of Observation 2 and Observation 3. If $k, r \in 4\mathbb{N}$ then i_{cx} coincides with the index i_{sb} in (4). If $k, r \in 4\mathbb{N} + 2$ and $m_1(X) < k(k+1)/4$ then there exist at least an index i_{cx} such that (17) holds. If instead, $k, r \in 4\mathbb{N} + 2$ and $m_1(X) > k(k+1)/4$ then there exist an index i_{cx} such that (16) holds. Also, note that if $k, r \in 4\mathbb{N} + 2$ then the definition of m_1-crossing index is stronger than the definition of simple m_1-balancing index as the following example shows.

Example 4 [Case $k = 6$, $r = 6$ and $p = 3$ (Continued)]: If $k = 6 \in 4\mathbb{N} + 2$ then $10 = \lfloor k(k+1)/4 \rfloor < \lfloor k(k+1)/4 \rfloor = 11$. The random walk sequence of the word $X = 010011 \in S(6, 3)$ is

$$\begin{aligned} X^{(d_0)} = X^{(\mathbf{0})} &= 010011, \quad m_1\left(X^{(\mathbf{0})}\right) = 13, \\ X^{(1)} &= \underline{10}0011, \quad m_1\left(X^{(1)}\right) = 12, \\ X^{(2)} &= 1\underline{00}011, \quad m_1\left(X^{(2)}\right) = 12, \\ X^{(3)} &= 10\underline{00}11, \quad m_1\left(X^{(3)}\right) = 12, \\ X^{(4)} &= 100\underline{10}1, \quad m_1\left(X^{(4)}\right) = 11, \\ X^{(5)} &= 1001\underline{10}, \quad m_1\left(X^{(5)}\right) = 10, \longleftarrow \\ X^{(6)} &= 100\underline{110}, \quad m_1\left(X^{(6)}\right) = 10, \\ X^{(7)} &= 10\underline{10}1\underline{0}, \quad m_1\left(X^{(7)}\right) = 9, \\ X^{(d_1)} = X^{(\mathbf{8})} &= 1\underline{10}01\underline{0}, \quad m_1\left(X^{(\mathbf{15})}\right) = 8, \\ X^{(9)} &= \underline{11}001\underline{0}, \quad m_1\left(X^{(9)}\right) = 8, \\ X^{(10)} &= \underline{101}01\underline{0}, \quad m_1\left(X^{(10)}\right) = 9, \\ X^{(11)} &= \underline{1}0\underline{01}1\underline{0}, \quad m_1\left(X^{(11)}\right) = 10, \\ X^{(12)} &= \underline{1}00\underline{110}, m_1\left(X^{(12)}\right) = 10, \\ X^{(d_2)} = X^{(\mathbf{13})} &= \underline{1}0\underline{1010}, \quad m_1\left(X^{(\mathbf{15})}\right) = 9, \\ X^{(14)} &= \underline{110}0\underline{10}, \quad m_1\left(X^{(14)}\right) = 8, \\ X^{(15)} &= \underline{110010}, m_1\left(X^{(13)}\right) = 8. \end{aligned}$$

Hence, X has only one m_1-crossing index $i_{cx} = 5$ (indicated by the arrow to the right) because (16) holds. In particular, $i_{cx} = 5$ is a decreasing m_1-crossing index for the random walk $RW(X)$.

Now we come to the main general theorem.

Theorem 2: Let $k, r \in 2\mathbb{N}$ be given so that $n \overset{\text{def}}{=} k + r \in 4\mathbb{N}$. Let $p \in \mathbb{N}$,

$$\mathcal{CS} = \{\Gamma_0, \Gamma_1, \dots, \Gamma_{p-1}\}$$

be a partition of a subset of $S(r, r/2)$ such that (8) holds for all $h \in [0, p-1]$ and let

$$\mathcal{B} = \{\langle\, \Gamma_h \rangle : \; h \in [0, p-1]\}$$

be the set of functions defined by (10) and (12). If

$$\frac{k(k-1)}{2} \le \sum_{h=0}^{p-1} |\Gamma_h| - \epsilon(k, r), \quad \textit{with}$$
$$\epsilon(k, r) \overset{\text{def}}{=} \begin{cases} 0 & \textit{if } k, r \in 4\mathbb{N}, \\ 1 & \textit{otherwise}, \end{cases} \quad (18)$$

then $\mathcal{B}$ is a set of m_1-balancing functions according to Definition 1. In particular, if we let

$$p = \left| S\left(r, r/2, \left\lfloor \frac{r(r+1)}{4} \right\rfloor\right)\right|,$$

$$\bigcup_{h=0}^{p-1} |\Gamma_h| = S(r, r/2)$$

and

$$\frac{k(k-1)}{2} \le \binom{r}{r/2} \tag{19}$$

then $\mathcal{B}$ is a set of m_1-balancing functions. Note that the inequality in (19) is the leftmost relation in (2).

Proof: First note that

$$\begin{aligned}\left\lceil \frac{k(k+1)}{4} \right\rceil + \left\lfloor \frac{r(r+1)}{4} \right\rfloor + \frac{kr}{2} &= \left\lfloor \frac{k(k+1)}{4} \right\rfloor + \left\lceil \frac{r(r+1)}{4} \right\rceil + \frac{kr}{2} \\ &= \frac{n(n+1)}{4}\end{aligned} \tag{20}$$

because $k, r \in 2\mathbb{N}$ and $n \in 4\mathbb{N}$.

Property 1) of Definition 1 follows because the family $\mathcal{CS}$ is a partition. With regard to the first moment relation in property 2) of Definition 1, given any $X \in S(k, k/2)$, we need to find/define an m_1-balancing index, $\hat{h} \in h_{bal}(X)$, for the proposed refined coding scheme in each of the cases which follow.

Case $k, r \in 4\mathbb{N}$. Let i_{cx} be an m_1-crossing index of X. For such index,

$$i_{cx} \in \left[0, \frac{k(k-1)}{2}\right) \text{ and } m_1\left(X^{(i_{cx})}\right) = \frac{k(k+1)}{4} \in \mathbb{N} \tag{21}$$

hold because of Definition 2. In this case, the integer intervals in (11) are given by

$$I_h \overset{\text{def}}{=} \left[d_h - \frac{|\Gamma_h| - 1}{2}, d_h + \frac{|\Gamma_h| - 1}{2}\right],$$

for all integer $h \in [0, p-1]$; and,

$$I_p \overset{\text{def}}{=} \left[\left\lfloor \frac{|\Gamma_0|}{2} \right\rfloor + \sum_{h=1}^{p-1} |\Gamma_h| + 1, \sum_{h=0}^{p-1} |\Gamma_h|\right). \tag{22}$$

From (18), the family $\{I_h : h \in [0, p]\}$ partitions the index range interval $[0, k(k-1)/2)$. So, there exists an integer $h \in [0, p]$ such that $i_{cx} \in I_h$. Exactly one of the following subcases must hold.

A1: $h \in [0, p-1]$. Here, from (22),

$$\delta i \overset{\text{def}}{=} i_{cx} - d_h \in \left[-\frac{|\Gamma_h| - 1}{2}, \frac{|\Gamma_h| - 1}{2}\right].$$

So, from (5),

$$\begin{aligned}m_1\left(X^{(d_h)}\right) &\in \left[m_1\left(X^{(i_{cx})}\right) - |\delta i|, m_1\left(X^{(i_{cx})}\right) + |\delta i|\right] \\ &= \left[\frac{k(k+1)}{4} - |\delta i|, \frac{k(k+1)}{4} + |\delta i|\right] \\ &\subseteq \left[\frac{k(k+1)}{4} - \frac{|\Gamma_h| - 1}{2}, \frac{k(k+1)}{4} + \frac{|\Gamma_h| - 1}{2}\right];\end{aligned}$$

and so, there always exists a "small m_1-imbalance"

$$\mu \in [-|\delta i|, |\delta i|] \cap \mathbb{N} \subseteq \left[-\frac{|\Gamma_h| - 1}{2}, \frac{|\Gamma_h| - 1}{2}\right]$$

such that,

$$\begin{aligned}m_1\left(X^{(d_h)}\right) = \frac{k(k+1)}{4} - \mu \in \Bigg[&\frac{k(k+1)}{4} - \frac{|\Gamma_h| - 1}{2}, \\ &\frac{k(k+1)}{4} + \frac{|\Gamma_h| - 1}{2}\Bigg].\end{aligned}$$

Hence, from (8), there exists $C \in \Gamma_h$ with

$$\begin{aligned}m_1(C) &= \frac{r(r+1)}{4} \\ &\quad + \mu \in \left[\frac{r(r+1)}{4} - \frac{|\Gamma_h| - 1}{2}, \frac{r(r+1)}{4} + \frac{|\Gamma_h| - 1}{2}\right] \\ &= m_1(\Gamma_h)\end{aligned}$$

such that

$$\begin{aligned}m_1(\langle \Gamma_h \rangle (X)\, C) &= m_1\left(X^{(d_h)}\right) + m_1(C) + km_0(C) \\ &= \left(\frac{k(k+1)}{4} - \mu\right) + \left(\frac{r(r+1)}{4} + \mu\right) + \frac{kr}{2} \\ &= \frac{n(n+1)}{4},\end{aligned}$$

because of (14) in Theorem 1 with

$$\begin{aligned}m_1(Y) &= m_1(X^{(d_h)}) = k(k+1)/4 - \mu, \\ m_1(C) &= r(r+1)/4 + \mu \quad \text{and} \\ km_0(C) &= kr/2;\end{aligned}$$

and also because of (20). This means that $\hat{h} \overset{\text{def}}{=} h \in h_{bal}(X)$ is a balancing index for this refined coding scheme.

A2: $h = p$. In this case, from (22) and Definition 2,

$$i_{cx} \in \left[\left\lfloor \frac{|\Gamma_0|}{2} \right\rfloor + \sum_{h=1}^{p-1} |\Gamma_h| + 1, \frac{k(k-1)}{2}\right).$$

So, from (18),

$$\begin{aligned}\frac{k(k-1)}{2} - i_{cx} &\in \left[1, \frac{k(k-1)}{2} - \left\lfloor \frac{|\Gamma_0|}{2} \right\rfloor - \sum_{h=1}^{p-1} |\Gamma_h|\right) \\ &\subseteq \left[1, \left\lceil \frac{|\Gamma_0|}{2} \right\rceil\right).\end{aligned}$$

Hence, from (5), (21) and the above relation,

$$\begin{aligned}m_1\left(X^R\right) &= m_1\left(X^{(k(k-1)/2)}\right) \\ &\in \Bigg[m_1\left(X^{(i_{cx})}\right) - \left(\frac{k(k-1)}{2} - i_{cx}\right), \\ &\qquad m_1\left(X^{(i_{cx})}\right) + \left(\frac{k(k-1)}{2} - i_{cx}\right)\Bigg] \\ &\subseteq \Bigg[\frac{k(k+1)}{4} - \left(\left\lceil \frac{|\Gamma_0|}{2} \right\rceil - 1\right), \\ &\qquad \frac{k(k+1)}{4} + \left(\left\lceil \frac{|\Gamma_0|}{2} \right\rceil - 1\right)\Bigg] \\ &= \left[\frac{k(k+1)}{4} - \frac{|\Gamma_0| - 1}{2}, \frac{k(k+1)}{4} + \frac{|\Gamma_0| - 1}{2}\right].\end{aligned}$$

This, from Observation 2, implies,

$$m_1(X) = \frac{k(k+1)}{2} - m_1\left(X^R\right) \in \left[\frac{k(k+1)}{4} - \frac{|\Gamma_0|-1}{2}, \frac{k(k+1)}{4} + \frac{|\Gamma_0|-1}{2}\right];$$

and so, there exists a "small m_1-imbalance" $\mu \in \mathbb{N}$ such that,

$$m_1(X) = \frac{k(k+1)}{4} - \mu \in \left[\frac{k(k+1)}{4} - \frac{|\Gamma_0|-1}{2}, \frac{k(k+1)}{4} + \frac{|\Gamma_0|-1}{2}\right].$$

Again, from (8), there exists $C \in \Gamma_0$ with

$$\begin{aligned} m_1(C) &= \frac{r(r+1)}{4} \\ &\quad +\mu \in \left[\frac{r(r+1)}{4} - \frac{|\Gamma_0|-1}{2}, \frac{r(r+1)}{4} + \frac{|\Gamma_0|-1}{2}\right] \\ &= m_1(\Gamma_0) \end{aligned}$$

such that $m_1(\langle \Gamma_0\rangle(X)\,C) = n(n+1)/4$ because of Theorem 1 and (20). So, if $i_{cx} \in I_p$ then $\hat{h} \overset{\text{def}}{=} 0 \in h_{bal}(X)$ is a balancing index for the refined coding scheme.

Case $k, r \in 4\mathbb{N} + 2$. Let

$$i_{cx} \in \left[0, \frac{k(k-1)}{2}\right] = \left[0, \frac{k(k-1)}{2} + 1\right)$$

be an m_1-crossing index of X defined in Definition 2. Here, the integer intervals in (11) are given by

$$I_h \overset{\text{def}}{=} \left[d_h - \frac{|\Gamma_h|}{2} + 1, d_h + \frac{|\Gamma_h|}{2}\right],$$

for all integer $h \in [0, p-1]$; and,

$$I_p \overset{\text{def}}{=} \left[\frac{|\Gamma_0|}{2} + \sum_{h=1}^{p-1} |\Gamma_h| + 1, \sum_{h=0}^{p-1} |\Gamma_h|\right). \tag{23}$$

From (18) and $\epsilon(k,r) = 1$, the family $\{I_h : h \in [0,p]\}$ partitions the index range interval $[0, k(k-1)/2]$. So, there exists an integer $h \in [0,p]$ such that $i_{cx} \in I_h$. Based on (23), exactly one of the following subcases must hold.

B1: $h \in [0, p-1]$, (16) holds and

$$\begin{aligned} i_{cx} = d_h + \frac{|\Gamma_h|}{2} \iff \delta i &\overset{\text{def}}{=} (i_{cx} - 1) - d_h \\ &= \frac{|\Gamma_h|}{2} - 1 = \frac{|\Gamma_h|-2}{2}. \end{aligned}$$

In this case, from (5) and (16), there always exists a "small m_1-imbalance" $\mu \in \mathbb{N}$ such that,

$$\begin{aligned} m_1\left(X^{(d_h)}\right) &= \left\lceil \frac{k(k+1)}{4} \right\rceil \\ &\quad -\mu \in \left[m_1\left(X^{(i_{cx}-1)}\right) - \delta i, m_1\left(X^{(i_{cx}-1)}\right) + \delta i\right] \\ &= \left[\left\lceil \frac{k(k+1)}{4} \right\rceil - \frac{|\Gamma_h|-2}{2}, \left\lceil \frac{k(k+1)}{4} \right\rceil + \frac{|\Gamma_h|-2}{2}\right]. \end{aligned}$$

Hence, from (8), there exists $C \in \Gamma_h$ with

$$\begin{aligned} m_1(C) &= \left\lfloor \frac{r(r+1)}{4} \right\rfloor \\ &\quad +\mu \in \left[\left\lfloor \frac{r(r+1)}{4} \right\rfloor - \frac{|\Gamma_h|-2}{2}, \left\lfloor \frac{r(r+1)}{4} \right\rfloor + \frac{|\Gamma_h|-2}{2}\right] \subseteq m_1(\Gamma_h) \end{aligned}$$

such that

$$\begin{aligned} &m_1(\langle \Gamma_h\rangle(X)\,C) \\ &= m_1\left(X^{(d_h)}\right) + m_1(C) + km_0(C) \\ &= \left(\left\lceil \frac{k(k+1)}{4} \right\rceil - \mu\right) + \left(\left\lfloor \frac{r(r+1)}{4} \right\rfloor + \mu\right) + \frac{kr}{2} \\ &= \frac{n(n+1)}{4}, \end{aligned}$$

because of Theorem 1 and (20). This means that $\hat{h} \overset{\text{def}}{=} h \in h_{bal}(X)$ is a balancing index for the refined coding scheme.

B2: $h \in [0, p-1]$, (16) holds and

$$\begin{aligned} &i_{cx} \in \left[d_h - \left(\frac{|\Gamma_h|}{2} - 1\right), d_h + \left(\frac{|\Gamma_h|}{2} - 1\right)\right] \\ &\iff \delta i \overset{\text{def}}{=} i_{cx} - d_h \in \left[-\frac{|\Gamma_h|-2}{2}, \frac{|\Gamma_h|-2}{2}\right]. \end{aligned}$$

In this case, from (5) and (16), there always exists a "small m_1-imbalance" $\mu \in \mathbb{N}$ such that,

$$\begin{aligned} m_1\left(X^{(d_h)}\right) &= \left\lfloor \frac{k(k+1)}{4} \right\rfloor \\ &\quad -\mu \in \left[m_1\left(X^{(i_{cx})}\right) - |\delta i|, m_1\left(X^{(i_{cx})}\right) + |\delta i|\right] \\ &= \left[\left\lfloor \frac{k(k+1)}{4} \right\rfloor - |\delta i|, \left\lfloor \frac{k(k+1)}{4} \right\rfloor + |\delta i|\right] \\ &\subseteq \left[\left\lfloor \frac{k(k+1)}{4} \right\rfloor - \frac{|\Gamma_h|-2}{2}, \left\lfloor \frac{k(k+1)}{4} \right\rfloor + \frac{|\Gamma_h|-2}{2}\right]. \end{aligned}$$

Hence, from (8), there exists $C \in \Gamma_h$ with

$$m_1(C) = \left\lceil \frac{r(r+1)}{4} \right\rceil + \mu \in \left[\left\lceil \frac{r(r+1)}{4} \right\rceil - \frac{|\Gamma_h|-2}{2}, \left\lceil \frac{r(r+1)}{4} \right\rceil + \frac{|\Gamma_h|-2}{2}\right] \subseteq m_1(\Gamma_h)$$

such that

$$\begin{aligned} &m_1(\langle \Gamma_h\rangle(X)\,C) \\ &= m_1\left(X^{(d_h)}\right) + m_1(C) + km_0(C) \\ &= \left(\left\lfloor \frac{k(k+1)}{4} \right\rfloor - \mu\right) + \left(\left\lceil \frac{r(r+1)}{4} \right\rceil + \mu\right) + \frac{kr}{2} \\ &= \frac{n(n+1)}{4}, \end{aligned}$$

because of Theorem 1 and (20). This means that $\hat{h} \overset{\text{def}}{=} h \in h_{bal}(X)$.

B3: $h = p$, (16) holds and

$$i_{cx} \in \left[\frac{|\Gamma_0|}{2} + \sum_{h=1}^{p-1} |\Gamma_h| + 1, \frac{k(k-1)}{2}\right].$$

So, from (18) and $\epsilon(k, r) = 1$,

$$\frac{k(k-1)}{2} - i_{cx} \in \left[0, \frac{k(k-1)}{2} - \frac{|\Gamma_0|}{2} - \sum_{h=1}^{p-1} |\Gamma_h|\right] \subseteq \left[0, \frac{|\Gamma_0|}{2} - 1\right].$$

Hence, from (5), (16) and the above relation,

$$\begin{aligned} m_1\left(X^R\right) &= m_1\left(X^{(k(k-1)/2)}\right) \\ &\in \left[m_1\left(X^{(i_{cx})}\right) - \left(\frac{k(k-1)}{2} - i_{cx}\right),\right. \\ &\qquad \left. m_1\left(X^{(i_{cx})}\right) + \left(\frac{k(k-1)}{2} - i_{cx}\right)\right] \\ &\subseteq \left[\left\lfloor \frac{k(k+1)}{4} \right\rfloor - \frac{|\Gamma_0| - 2}{2},\right. \\ &\qquad \left. \left\lfloor \frac{k(k+1)}{4} \right\rfloor + \frac{|\Gamma_0| - 2}{2}\right] \end{aligned}$$

This, from Observation 2, implies,

$$m_1(X) = \frac{k(k+1)}{2} - m_1\left(X^R\right) \in \left[\left\lceil \frac{k(k+1)}{4} \right\rceil - \frac{|\Gamma_0| - 2}{2}, \left\lceil \frac{k(k+1)}{4} \right\rceil + \frac{|\Gamma_0| - 2}{2}\right];$$

and so, there exists a "small m_1-imbalance" $\mu \in \mathbb{N}$ such that,

$$m_1(X) = \left\lceil \frac{k(k+1)}{4} \right\rceil - \mu \in \left[\left\lceil \frac{k(k+1)}{4} \right\rceil - \frac{|\Gamma_0| - 2}{2}, \left\lceil \frac{k(k+1)}{4} \right\rceil + \frac{|\Gamma_0| - 2}{2}\right].$$

Hence, from (8), there exists $C \in \Gamma_0$ with

$$m_1(C) = \left\lfloor \frac{r(r+1)}{4} \right\rfloor + \mu \in \left[\left\lfloor \frac{r(r+1)}{4} \right\rfloor - \frac{|\Gamma_0| - 2}{2}, \left\lfloor \frac{r(r+1)}{4} \right\rfloor + \frac{|\Gamma_0| - 2}{2}\right] \subseteq m_1(\Gamma_0)$$

such that $m_1(\langle \Gamma_0 \rangle(X)\, C) = n(n+1)/4$ because of Theorem 1 and (20). So, if $i_{cx} \in I_p$ then $\hat{h} \overset{\text{def}}{=} 0 \in h_{bal}(X)$.

C1: $h \in [0, p-1]$, (17) holds and

$$i_{cx} = d_h + \frac{|\Gamma_h|}{2} \iff \delta i \overset{\text{def}}{=} (i_{cx} - 1) - d_h] = \frac{|\Gamma_h|}{2} - 1 = \frac{|\Gamma_h| - 2}{2}.$$

This case follows exactly as the case **B1** where the floor function is exchanged with the ceiling function. Here $\hat{h} \overset{\text{def}}{=} h \in h_{bal}(X)$.

C2: $h \in [0, p-1]$, (17) holds and

$$i_{cx} \in \left[d_h - \left(\frac{|\Gamma_h|}{2} - 1\right), d_h + \left(\frac{|\Gamma_h|}{2} - 1\right)\right] \iff \delta i \overset{\text{def}}{=} i_{cx} - d_h \in \left[-\frac{|\Gamma_h| - 2}{2}, \frac{|\Gamma_h| - 2}{2}\right].$$

This case follows exactly as the case **B2** where the floor function is exchanged with the ceiling function. Here $\hat{h} \overset{\text{def}}{=} h \in h_{bal}(X)$.

C3: $h = p$, (17) holds and

$$i_{cx} \in \left[\frac{|\Gamma_0|}{2} + \sum_{h=1}^{p-1} |\Gamma_h| + 1, \frac{k(k-1)}{2}\right].$$

This case follows exactly as the case **B3** where the floor function is exchanged with the ceiling function. Here $\hat{h} \overset{\text{def}}{=} 0 \in h_{bal}(X)$.

Since no other cases are left, property 2) holds. As we can see, property 2) follows similarly to the proof of [19, Th. 4]. In fact, here we have even proven a slightly stronger result; namely, we have shown that if an m_1-crossing index $i_{cx} \in [0, k(k-1)/2 + [(k/2) \bmod 2])$ of the random walk $RW(X)$ in (15) lies in I_h then $h \in h_{bal}(X) \in [0, p-1]$ is a balancing index for the refined scheme. This, modulus the necessary length of the random walk given by $k(k-1)/2 + [(k/2) \bmod 2]$ (see case A2, B3 and C3 above). In Section III, we will make a good use of this improved result.

Property 3) of Definition 1 holds because the functions $\langle \Gamma_h \rangle$ in (12) are permutations of the components of X. So, $\mathcal{B}$ is a set of m_1-balancing functions according to Definition 1. In particular, this statement holds true if we let $\bigcup_{h=0}^{p-1} |\Gamma_h| = S(r, r/2)$. In this case, k must satisfy

$$\frac{k(k-1)}{2} \le |S(r, r/2)| - \epsilon(k, r);$$

which is equivalent to $k(k-1)/2 \le \binom{r}{r/2}$ because $k, r \in 2\mathbb{N}$, $n \in 4\mathbb{N}$ and because whenever $k, r \in 4\mathbb{N} + 2$ it follows $k(k-1)/2 \in 2\mathbb{N} + 1$ and $\binom{r}{r/2} \in 2\mathbb{N}$. ■

At this point, the encoding of an information word is done as follows.

Algorithm 1 General Encoding Algorithm

Input: the information word $Z \in \mathbb{Z}_2^{\tilde{k}}$.

Output: the codeword

$$Y\,C = \mathcal{E}_2(\mathcal{E}_1(Z)) = (\mathcal{E}_1 \circ \mathcal{E}_2)(Z) \in \mathcal{SN}(n, 2)$$

where $n = k + r$, $Y \in \mathbb{Z}_2^k$ and $C \in \mathbb{Z}_2^r$.

Perform steps S0, S1 and S2.

S0: m_0-balance the information word Z. For example, any m_0-balancing method given in [1], [2], [4], [7], [12], [18], [19] or [21] can be used. Let X be the m_0-balanced information word of length k associated with Z; i. e., $X = \mathcal{E}_1(Z)$. In this way, $m_0(X) = k/2$.

S1: find a balancing index of X and then compute the relative information and check part of the codeword. For the time being, assume this is done exhaustively by computing the set, $h_{bal}(X)$, of all possible balancing indices associated with X as follows. For all $h \in [0, p-1]$ do steps S1.1, S1.2, S1.3 and S1.4.

S1.1: compute $\langle \Gamma_h \rangle(X) = X^{(d_h)}$.

S1.2: compute $m_1(\langle \Gamma_h \rangle(X))$.

S1.3: compute the m_1-unbalance

$$\mu_1 = \frac{n(n+1)}{4} - m_1\left(\langle \Gamma_h \rangle (X)\right) - \frac{kr}{2}. \tag{24}$$

S1.4: if there exists $C \in \Gamma_h$ such that $m_1(C) = \mu_1$ then a balancing index $h_{bal} \overset{\text{def}}{=} h$ is found and the relative m_1-balancing check is $C_{h_{bal}} \overset{\text{def}}{=} C \in \Gamma_{h_{bal}}$. Note that such m_1-balancing check exists in Γ_h if, and only if, $\mu_1 \in m_1(\Gamma_h) = [\alpha_h, \beta_h]$ (because of Observation 4 above).
S2: for the m_0-balanced information word X, select the m_1-balanced codeword as

$$\mathcal{E}_2(X) = Y\,C = \left\langle \Gamma_{h_{bal}} \right\rangle (X)\, C_{h_{bal}} = X^{(d_{h_{bal}})}\, C_{h_{bal}};$$

where $h_{bal} \in h_{bal}(X) \subseteq [0, p-1]$ is one among all possible balancing indices found in step S1. In this way, the information word Z is encoded into the 2-OSN word $Y\,C = \mathcal{E}_2(X) = \mathcal{E}_2(\mathcal{E}_1(Z))$.
S3: Output $\mathcal{E}_2(X)$ and **exit**.

On the other hand, the decoding is performed as follows.

```
Algorithm 2 General Decoding Algorithm
Input: the codeword associated with some $Z \in \mathbb{Z}_2^{\tilde{k}}$,
    $Y\,C = \mathcal{E}_2(\mathcal{E}_1(Z)) = (\mathcal{E}_1 \circ \mathcal{E}_2)(Z) \in \mathcal{SN}(n, 2)$.
with $n = k + r$, $Y \in \mathbb{Z}_2^k$ and $C \in \mathbb{Z}_2^r$.
Output: the information word $Z = (\mathcal{E}_1 \circ \mathcal{E}_2)^{-1}(Y\,C)$.
Perform steps S0, S1 and S2.
S0: compute the index $h_{bal} \in [0, p-1]$ such that $C \in \Gamma_{h_{bal}}$.
S1: compute the word $X = \mathcal{E}_2^{-1}(Y\,C) = \langle \Gamma_{h_{bal}} \rangle^{-1}(Y)$.
S2: undo the $m_0$-balancing of the word $X$ using the method chosen in step S0 of Algorithm 1. Let $Z = \mathcal{E}_1^{-1}(X) \in \mathbb{Z}_2^{\tilde{k}}$ be the information word whose $m_0$-balanced encoding is $X$. In this way, the word $Y\,C$ is decoded into the word
    $Z = \mathcal{E}_1^{-1}(X) = \mathcal{E}_1^{-1}(\mathcal{E}_2^{-1}(Y\,C)) = (\mathcal{E}_1 \circ \mathcal{E}_2)^{-1}(Y\,C)$.
S3: Output $Z$ and exit.
```

The following example shows the m_1-balancing part of Algorithm 1 and the m_1-unbalancing part of Algorithm 2.

Example 5 (Code Design for $k = 12$, $r = 8$ and $p = 8$): Let the number of m_0-balanced information bits be $k = 12 \in 2\mathbb{N}$. So, if $r = 8 \in 2\mathbb{N}$ then relation (19) holds because

$$\frac{k(k-1)}{2} = 66 \le 70 = \binom{8}{4} = \binom{r}{r/2}.$$

This means that with $r = 8$, by letting

$$\bigcup_{h=0}^{p-1} |\Gamma_h| = S(8, 4),$$

a code design of length $n = k + r = 20$ with

$$p = s\left(8, 4, 8 \cdot \frac{9}{4} = 18\right) = 8$$

m_1-balancing functions can be given to m_1-balance the $k = 12$ m_0-balanced bits. Columns 2 through 8 of Table I define a particular partition $\mathcal{CS} = \{\Gamma_0, \Gamma_1, \ldots, \Gamma_7\}$. We readily note that Γ_7 is redundant and a code design can be given with $p = 7$ m_1-balancing functions by using the first 7 Γ_h. This is because

$$\frac{k(k-1)}{2} = 66 \le 69 = 70 - |\Gamma_7| = \sum_{h=0}^{6} |\Gamma_h|$$

and so (18) is satisfied. However, note that by using $p = 8$, instead of 7, m_1-balancing functions we increase the average number of possible encodings of a given information word. Thus, by applying the same method given in [12], [16], and [29] for balanced codes to the 2-OSN codes given here, a little more extra information can be conveyed by the code. Now, the m_0-weight of every codeword must be equal to $n/2 = 10$ and the m_1-weight must be equal to $n(n+1)/4 = 105$. Suppose

$$X = 100000101111$$

is the given m_0-balanced word which needs to be encoded into a word of $\mathcal{SN}(20, 2)$. The entire random walk sequence for X is given below (where the left arrows to the right indicate a word in $\mathcal{SN}(12, 2)$).

$$\begin{aligned}
X^{(d_0)} = X^{(\mathbf{0})} &= 100000101111, & m_1\left(X^{(\mathbf{0})}\right) &= 50,\\
X^{(1)} &= \underline{01}0000101111, & m_1\left(X^{(1)}\right) &= 51,\\
X^{(2)} &= 0\underline{01}000101111, & m_1\left(X^{(2)}\right) &= 52,\\
X^{(3)} &= 00\underline{01}00101111, & m_1\left(X^{(3)}\right) &= 53,\\
X^{(4)} &= 000\underline{01}0101111, & m_1\left(X^{(4)}\right) &= 54,\\
X^{(5)} &= 0000\underline{01}101111, & m_1\left(X^{(5)}\right) &= 55,\\
X^{(6)} &= 00000\underline{11}01111, & m_1\left(X^{(6)}\right) &= 55,\\
X^{(7)} &= 000001\underline{01}1111, & m_1\left(X^{(7)}\right) &= 56,\\
X^{(8)} &= 0000010\underline{11}111, & m_1\left(X^{(8)}\right) &= 56,\\
X^{(9)} &= 00000101\underline{11}11, & m_1\left(X^{(9)}\right) &= 56,\\
X^{(10)} &= 000001011\underline{11}1, & m_1\left(X^{(10)}\right) &= 56,\\
X^{(11)} &= 0000010111\underline{1}\underline{\underline{1}}, & m_1\left(X^{(11)}\right) &= 56,\\
X^{(12)} &= 000001011\underline{11}\underline{\underline{1}}, & m_1\left(X^{(12)}\right) &= 56,\\
X^{(13)} &= 00000101\underline{11}1\underline{\underline{1}}, & m_1\left(X^{(13)}\right) &= 56,\\
X^{(14)} &= 0000010\underline{11}11\underline{\underline{1}}, & m_1\left(X^{(14)}\right) &= 56,\\
X^{(d_1)} = X^{(\mathbf{15})} &= 000001\underline{10}111\underline{\underline{1}}, & m_1\left(X^{(\mathbf{15})}\right) &= 55,\\
X^{(16)} &= 00000\underline{11}0111\underline{\underline{1}}, & m_1\left(X^{(16)}\right) &= 55,\\
X^{(17)} &= 0000\underline{10}10111\underline{\underline{1}}, & m_1\left(X^{(17)}\right) &= 54,\\
X^{(18)} &= 000\underline{10}010111\underline{\underline{1}}, & m_1\left(X^{(18)}\right) &= 53,
\end{aligned}$$

$$
\begin{aligned}
X^{(19)} &= 00\underline{10}00010111\underline{1}, \quad m_1\left(X^{(19)}\right) = 52,\\
X^{(20)} &= 0\underline{10}00010111\underline{1}, \quad m_1\left(X^{(20)}\right) = 51,\\
X^{(21)} &= \underline{10}000010111\underline{1}, \quad m_1\left(X^{(21)}\right) = 50,\\
X^{(22)} &= \underline{100}00010111\underline{1}, \quad m_1\left(X^{(22)}\right) = 50,\\
X^{(23)} &= \underline{1}0\underline{00}0010111\underline{1}, \quad m_1\left(X^{(23)}\right) = 50,\\
X^{(24)} &= \underline{1}00\underline{00}010111\underline{1}, \quad m_1\left(X^{(24)}\right) = 50,\\
X^{(25)} &= \underline{1}000\underline{00}10111\underline{1}, \quad m_1\left(X^{(25)}\right) = 50,\\
X^{(26)} &= \underline{1}0000\underline{10}0111\underline{1}, \quad m_1\left(X^{(26)}\right) = 49,\\
X^{(d_2)} = X^{(\mathbf{27})} &= \underline{1}00001\underline{00}111\underline{1}, \quad m_1\left(X^{(\mathbf{27})}\right) = 49,\\
X^{(28)} &= \underline{1}000010\underline{10}11\underline{1}, \quad m_1\left(X^{(28)}\right) = 48,\\
X^{(29)} &= \underline{1}0000101\underline{10}1\underline{1}, \quad m_1\left(X^{(29)}\right) = 47,\\
X^{(30)} &= \underline{1}00001011\underline{101}, \quad m_1\left(X^{(30)}\right) = 46,\\
X^{(31)} &= \underline{1}0000101\underline{1101}, \quad m_1\left(X^{(31)}\right) = 46,\\
X^{(32)} &= \underline{1}000010\underline{11}1\underline{01}, \quad m_1\left(X^{(32)}\right) = 46,\\
X^{(33)} &= \underline{1}00001\underline{10}11\underline{01}, \quad m_1\left(X^{(33)}\right) = 45,\\
X^{(34)} &= \underline{1}0000\underline{11}011\underline{01}, \quad m_1\left(X^{(34)}\right) = 45,\\
X^{(35)} &= \underline{1}000\underline{10}1011\underline{01}, \quad m_1\left(X^{(35)}\right) = 44,\\
X^{(36)} &= \underline{1}00\underline{10}01011\underline{01}, \quad m_1\left(X^{(36)}\right) = 43,\\
X^{(38)} &= \underline{110}0001011\underline{01}, \quad m_1\left(X^{(38)}\right) = 41,\\
X^{(d_3)} = X^{(\mathbf{37})} &= \underline{1}0\underline{10}001011\underline{01}, \quad m_1\left(X^{(\mathbf{37})}\right) = 42,\\
X^{(39)} &= \underline{1100}001011\underline{01}, \quad m_1\left(X^{(39)}\right) = 41,\\
X^{(40)} &= \underline{11}0\underline{00}01011\underline{01}, \quad m_1\left(X^{(40)}\right) = 41,\\
X^{(41)} &= \underline{11}00\underline{00}1011\underline{01}, \quad m_1\left(X^{(41)}\right) = 41,\\
X^{(42)} &= \underline{11}000\underline{10}011\underline{01}, \quad m_1\left(X^{(42)}\right) = 40,\\
X^{(43)} &= \underline{11}0001\underline{00}11\underline{01}, \quad m_1\left(X^{(43)}\right) = 40,\\
X^{(44)} &= \underline{11}00010\underline{10}1\underline{01}, \quad m_1\left(X^{(44)}\right) = 39, \longleftarrow\\
X^{(45)} &= \underline{11}000101\underline{1001}, \quad m_1\left(X^{(45)}\right) = 38,\\
X^{(d_4)} = X^{(\mathbf{46})} &= \underline{11}00010\underline{11001}, \quad m_1\left(X^{(\mathbf{46})}\right) = 38,\\
X^{(47)} &= \underline{11}0001\underline{10}1\underline{001}, \quad m_1\left(X^{(47)}\right) = 37,\\
X^{(48)} &= \underline{11}000\underline{11}01\underline{001}, \quad m_1\left(X^{(48)}\right) = 37,\\
X^{(49)} &= \underline{11}00\underline{10}101\underline{001}, \quad m_1\left(X^{(49)}\right) = 36,\\
X^{(50)} &= \underline{11}0\underline{10}0101\underline{001}, \quad m_1\left(X^{(50)}\right) = 35,\\
X^{(51)} &= \underline{1110}00101\underline{001}, \quad m_1\left(X^{(51)}\right) = 34,\\
X^{(52)} &= \underline{11100}0101\underline{001}, \quad m_1\left(X^{(52)}\right) = 34,
\end{aligned}
$$

$$
\begin{aligned}
X^{(d_5)} = X^{(\mathbf{53})} &= \underline{111}0\underline{00}101\underline{001}, \quad m_1\left(X^{(\mathbf{53})}\right) = 34,\\
X^{(54)} &= \underline{111}00\underline{10}01\underline{001}, \quad m_1\left(X^{(54)}\right) = 33,\\
X^{(55)} &= \underline{111}001\underline{00}1\underline{001}, \quad m_1\left(X^{(55)}\right) = 33,\\
X^{(56)} &= \underline{111}0010\underline{10001}, \quad m_1\left(X^{(56)}\right) = 32,\\
X^{(57)} &= \underline{111}001\underline{100001}, \quad m_1\left(X^{(57)}\right) = 31,\\
X^{(d_6)} = X^{(\mathbf{58})} &= \underline{111}00\underline{11}\underline{00001}, \quad m_1\left(X^{(\mathbf{58})}\right) = 31,\\
X^{(59)} &= \underline{111}0\underline{10}10\underline{0001}, \quad m_1\left(X^{(59)}\right) = 30,\\
X^{(60)} &= \underline{11110}010\underline{0001}, \quad m_1\left(X^{(60)}\right) = 29,\\
X^{(d_7)} = X^{(\mathbf{61})} &= \underline{111100}10\underline{0001}, \quad m_1\left(X^{(\mathbf{61})}\right) = 29,\\
X^{(62)} &= \underline{1111}0\underline{10}0\underline{0001}, \quad m_1\left(X^{(62)}\right) = 28,\\
X^{(63)} &= \underline{1111}01\underline{000001}, \quad m_1\left(X^{(63)}\right) = 28,\\
X^{(64)} &= \underline{1111}00\underline{100001}, \quad m_1\left(X^{(64)}\right) = 29,\\
X^{(65)} &= \underline{111100}1\underline{00001}, \quad m_1\left(X^{(65)}\right) = 29,\\
X^{(66)} &= \underline{111101000001}, \quad m_1\left(X^{(66)}\right) = 28.
\end{aligned}
$$

Note that, unlike the simple encoding scheme in [20] where all the $k(k-1)/2 = 66$ *possible functions* $X \to X^{(i)}$, $i \in [0, k(k-1)/2)$, *are used as the* m_1*-balancing functions, here only* $p = 8$ *of them are used. In fact, the family of the* m_1*-balancing functions* $\mathcal{B} = \{\langle \Gamma_h \rangle : h \in [0,7]\}$ *is defined as follows. From the definition of* Γ_h *in Table I, along with (10) and (12), we have* $\langle \Gamma_h \rangle(X) = X^{(d_h)}$, $h \in [0,7]$, *with* $d_0 = 0$, $d_1 = 15$, $d_2 = 27$, $d_3 = 37$, $d_4 = 46$, $d_5 = 53$, $d_6 = 58$ *and* $d_7 = 61$. *That is,*

$$
\begin{aligned}
\langle \Gamma_0 \rangle(X) &= X^{(\mathbf{0})} = x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}x_{11}x_{12} \overset{\text{def}}{=} X,\\
\langle \Gamma_1 \rangle(X) &= X^{(\mathbf{15})} = x_2x_3x_4x_5x_6x_7\underline{x_{12}x_8}x_9x_{10}x_{11}\underline{x_1},\\
\langle \Gamma_2 \rangle(X) &= X^{(\mathbf{27})} = \underline{x_{12}}x_3x_4x_5x_6x_7\underline{x_8x_2}x_9x_{10}x_{11}\underline{x_1},\\
\langle \Gamma_3 \rangle(X) &= X^{(\mathbf{37})} = \underline{x_{12}}x_3\underline{x_{11}x_4}x_5x_6x_7x_8x_9x_{10}\underline{x_2x_1},\\
\langle \Gamma_4 \rangle(X) &= X^{(\mathbf{46})} = \underline{x_{12}x_{11}}x_4x_5x_6x_7x_8\underline{x_{10}x_9x_3x_2x_1},\\
\langle \Gamma_5 \rangle(X) &= X^{(\mathbf{53})} = \underline{x_{12}x_{11}x_{10}}x_5\underline{x_6x_4}x_7x_8x_9\underline{x_3x_2x_1},\\
\langle \Gamma_6 \rangle(X) &= X^{(\mathbf{58})} = \underline{x_{12}x_{11}x_{10}}x_5x_6\underline{x_9x_7}x_8\underline{x_4x_3x_2x_1},\\
\langle \Gamma_7 \rangle(X) &= X^{(\mathbf{61})} = \underline{x_{12}x_{11}x_{10}x_9x_6x_5}x_7x_8\underline{x_4x_3x_2x_1}.
\end{aligned}
$$

In this way, letting

$$M_1 \overset{\text{def}}{=} \frac{n(n+1)}{4} - \frac{kr}{2} = 105 - 48 = 57,$$

the encoding Algorithm 1 executes steps S1 and S2 as follows.
S1 *(for* $h = 0$ *and* $d_0 = 0$*): compute*

$$
\begin{aligned}
\langle \Gamma_0 \rangle(X) &= X^{(0)} = 100000101111 (= X),\\
m_1\left(X^{(0)}\right) &= 50,\\
\mu_1 &= M_1 - m_1\left(X^{(0)}\right) = 57 - 50 = 7.
\end{aligned}
$$

We have $\mu_1 = 7 \notin m_1(\Gamma_0) = [10, 26] \iff$ *there is no check word* $C \in \Gamma_0$ *such that* $m_1(C) = \mu_1$.

TABLE I

EXAMPLE WITH $n = 20$, $k = 12$, $r = 8$ AND $p = 8$

$S(8,4,\mu) \setminus \Gamma_h$	Γ_0	Γ_1	Γ_2	Γ_3	Γ_4	Γ_5	Γ_6	Γ_7	$s(8,4,\mu)$
$S(8,4,\alpha=10) =$	11110000 $\boxed{}$								1
$S(8,4,11) =$	11101000 $\boxed{}$								1
$S(8,4,12) =$	11011000 $\boxed{}$	11100100 $\boxed{9}$							2
$S(8,4,13) =$	10111000 $\boxed{}$	11010100 $\boxed{10}$	11100010 $\boxed{22}$						3
$S(8,4,14) =$	01111000 $\boxed{62}$	10110100 $\boxed{11}$	11001100 $\boxed{23}$	11010010 $\boxed{33}$	11100001 $\boxed{42}$				5
$S(8,4,15) =$	01110100 $\boxed{63}$	10101100 $\boxed{12}$	10110010 $\boxed{24}$	11001010 $\boxed{34}$	11010001 $\boxed{43}$				5
$S(8,4,16) =$	01101100 $\boxed{64}$	10011100 $\boxed{13}$	01110010 $\boxed{25}$	10101010 $\boxed{35}$	11000110 $\boxed{44}$	10110001 $\boxed{51}$	11001001 $\boxed{56}$		7
$S(8,4,17) =$	01011100 $\boxed{65}$	01101010 $\boxed{14}$	10011010 $\boxed{26}$	10100110 $\boxed{36}$	01110001 $\boxed{45}$	10101001 $\boxed{52}$	11000101 $\boxed{57}$		7
$S(8,4,18) =$	00111100 $\boxed{0}$	01011010 $\boxed{15}$	01100110 $\boxed{27}$	10010110 $\boxed{37}$	01101001 $\boxed{46}$	10011001 $\boxed{53}$	10100101 $\boxed{58}$	11000011 $\boxed{61}$	8
$S(8,4,19) =$	00111010 $\boxed{1}$	01010110 $\boxed{16}$	10001110 $\boxed{28}$	01011001 $\boxed{38}$	01100101 $\boxed{47}$	10010101 $\boxed{54}$	10100011 $\boxed{59}$		7
$S(8,4,20) =$	00110110 $\boxed{2}$	01001110 $\boxed{17}$	00111001 $\boxed{29}$	01010101 $\boxed{39}$	10001101 $\boxed{48}$	01100011 $\boxed{55}$	10010011 $\boxed{60}$		7
$S(8,4,21) =$	00101110 $\boxed{3}$	00110101 $\boxed{18}$	01001101 $\boxed{30}$	01010011 $\boxed{40}$	10001011 $\boxed{49}$				5
$S(8,4,22) =$	00011110 $\boxed{4}$	00101101 $\boxed{19}$	00110011 $\boxed{31}$	01001011 $\boxed{41}$	10000111 $\boxed{50}$				5
$S(8,4,23) =$	00011101 $\boxed{5}$	00101011 $\boxed{20}$	01000111 $\boxed{32}$						3
$S(8,4,24) =$	00011011 $\boxed{6}$	00100111 $\boxed{21}$							2
$S(8,4,25) =$	00010111 $\boxed{7}$								1
$S(8,4,\beta=26) =$	00001111 $\boxed{8}$								1
$\lvert\Gamma_h\rvert =$	17	13	11	9	9	5	5	1	70
$d_h =$	0	15	27	37	46	53	58	61	$p = 8$

For any integer $\mu \in [10, 26]$, the $(\mu - 8)$th row gives the vectors $C = c_1 c_2 \ldots c_8 \in S(8,4,\mu)$ in ascending lexicographic order where c_1 is the least significant symbol, c_8 is the most significant symbol and $0 < 1$. The last column give $s(8,4,\mu) = |S(8,4,\mu)|$ in (9), for all integers $\mu \in [10, 26]$. For any integer $h \in [0, 7]$, the $(h + 2)$th column gives Γ_h. The 19th and 20th rows give $|\Gamma_h|$ and d_h, for all $h \in [0, 7]$, respectively. For all integers $\mu \in [10, 26]$ and $h \in [0, 7]$, if the entry in the lower right corner of the cell in the $(\mu - 8)$th row and $(h + 2)$th column is the integer $i \in [0, 65]$ then $\hat{h}(i) = h$, where $\hat{h} : [0, 65] \cap \mathbb{N} \to [0, 7]$ is the function in (25).

S1 *(for $h = 1$ and $d_1 = 15$): compute*

$$\langle \Gamma_1 \rangle(X) = X^{(15)} = 000001\underline{10}111\underline{1},$$
$$m_1\left(X^{(15)}\right) = 55,$$
$$\mu_1 = M_1 - m_1\left(X^{(15)}\right) = 57 - 55 = 2.$$

We have $\mu_1 = 2 \notin m_1(\Gamma_1) = [12, 24] \iff$ there is no check word $C \in \Gamma_1$ such that $m_1(C) = \mu_1$.

S1 *(for $h = 2$ and $d_2 = 27$): compute*

$$\langle \Gamma_2 \rangle(X) = X^{(27)} = \underline{1}00001\underline{00}111\underline{1},$$
$$m_1\left(X^{(27)}\right) = 49,$$
$$\mu_1 = M_1 - m_1\left(X^{(27)}\right) = 57 - 49 = 8.$$

We have $\mu_1 = 8 \notin m_1(\Gamma_2) = [13, 23] \iff$ there is no check word $C \in \Gamma_2$ such that $m_1(C) = \mu_1$.

S1 *(for $h = 3$ and $d_3 = 37$): compute*

$$\langle \Gamma_3 \rangle(X) = X^{(37)} = \underline{1}0\underline{10}0010110\underline{1},$$
$$m_1\left(X^{(37)}\right) = 42,$$
$$\mu_1 = M_1 - m_1\left(X^{(37)}\right) = 57 - 42 = 15.$$

The check word $C = 11001010 \in \Gamma_3$ is such that $m_1(C) = \mu_1 = 15 \in m_1(\Gamma_3) = [14, 22]$. So, $h_{bal} \overset{\text{def}}{=} 3$ is an m_1-balancing index and the relative m_1-balancing check is $C_3 \overset{\text{def}}{=} 11001010 \in \Gamma_3$.

S1 *(for $h = 4$ and $d_4 = 46$): compute*

$$\langle \Gamma_4 \rangle(X) = X^{(46)} = \underline{11}00010\underline{11001},$$
$$m_1\left(X^{(46)}\right) = 38,$$
$$\mu_1 = M_1 - m_1\left(X^{(46)}\right) = 57 - 38 = 19.$$

The check word $C = 01100101 \in \Gamma_4$ is such that $m_1(C) = \mu_1 = 19 \in m_1(\Gamma_4) = [14, 22]$. So, $h_{bal} \overset{\text{def}}{=} 4$ is an m_1-balancing index and the relative m_1-balancing check is $C_4 \overset{\text{def}}{=} 01100101 \in \Gamma_4$.

S1 *(for $h = 5$ and $d_5 = 53$): compute*

$$fun\Gamma_5(X) = X^{(53)} = \underline{111}0\underline{00}101\underline{001},$$
$$m_1\left(X^{(53)}\right) = 34,$$
$$\mu_1 = M_1 - m_1\left(X^{(53)}\right) = 57 - 34 = 23.$$

We have $\mu_1 = 23 \notin m_1(\Gamma_5) = [16, 20] \iff$ there is no check word $C \in \Gamma_5$ such that $m_1(C) = \mu_1$.

TABLE II
COMPARISONS WITH THE OPTIMAL CODES AND THE CODES IN [20] FOR SOME CODE LENGTH VALUES $n \in 4\mathbb{N}$

n	optimal	new scheme			scheme in [20]		
	$\tilde{k}_{op}$	k	$\tilde{k}$	$\Delta\tilde{k}_{op}$	$k_{[20]}$	$\tilde{k}_{[20]}$	$\Delta\tilde{k}_{[20]}$
4	1	2	1	0	–	1	0
8	3	4	2	1	–	3	−1
12	5	6	4	1	–	5	−1
16	9	8	6	3	–	9	−3
20	12	12	9	3	–	12	−3
24	15	14	11	4	–	15	−4
28	19	18	15	4	12	9	6
32	23	22	19	4	16	13	6
36	26	24	21	5	20	17	4
40	30	28	25	5	24	21	4
44	34	32	29	5	28	25	4
48	37	36	33	4	32	29	4
52	41	40	37	4	32	29	8
56	45	42	38	7	36	33	5
60	49	46	42	7	40	37	5
64	53	50	46	7	44	40	6
128	115	112	108	7	104	100	8
256	241	238	233	8	232	227	6
512	495	492	487	8	476	471	16
1024	1005	1002	996	9	988	982	14
2048	2027	2024	2018	9	2008	2002	16
4096	4073	4070	4063	10	4056	4049	14
8192	8167	8164	8157	10	8148	8141	16
16384	16357	16354	16346	11	16340	16332	14
32768	32739	32736	32728	11	32720	32712	16
65536	65505	65502	65493	12	65488	65479	14

The first column is the overall length, $n \in 4\mathbb{N}$, of 2-OSN codes. The second column gives the information bits for the optimal 2-OSN code of length n. The third, fourth and fifth refer to the new coding scheme given in this paper. The sixth, seventh and eight refer to the coding scheme given in [20]. In particular, for each coding scheme, k is the maximum number of m_0-balanced bits that can be m_1-balanced into n bits and $\tilde{k}$ is the maximum number of information bits that can be m_0-balanced into k bits. In the fifth column, $\Delta\tilde{k}_{op} = \tilde{k}_{op} - \tilde{k}$. In the eigth column, $\Delta\tilde{k}_{[20]} = \tilde{k} - \tilde{k}_{[20]}$. For $n \geq 128$, the values of the second column are obtained with the approximation $|\mathcal{SN}(n,2)| \approx \left\lfloor (4\sqrt{3}/\pi)2^n/n^2 \right\rfloor$ given in [20], [22].

S1 *(for $h = 6$ and $d_5 = 58$): compute*

$$\langle\,\Gamma_6\rangle(X) = X^{(58)} = \underline{111}00\underline{11}0\underline{0001},$$
$$m_1\left(X^{(58)}\right) = 31,$$
$$\mu_1 = M_1 - m_1\left(X^{(58)}\right) = 57 - 31 = 26.$$

We have $\mu_1 = 26 \notin m_1(\Gamma_6) = [16, 20] \iff$ there is no check word $C \in \Gamma_6$ such that $m_1(C) = \mu_1$.

S1 *(for $h = 7$ and $d_5 = 61$): compute*

$$\langle\,\Gamma_7\rangle(X) = X^{(61)} = \underline{111100}10\underline{0001},$$
$$m_1\left(X^{(61)}\right) = 29,$$
$$\mu_1 = M_1 - m_1\left(X^{(58)}\right) = 57 - 29 = 28.$$

We have $\mu_1 = 28 \notin m_1(\Gamma_7) = [18, 18] \iff$ there is no check word $C \in \Gamma_7$ such that $m_1(C) = \mu_1$.

At this point, execute step S2.

S2*: For example, select as encoding of the m_0-balanced word X the m_1-balanced word*

$$\mathcal{E}_2(X) = X^{(37)}\,C_3 = \underline{1}0\underline{10}001011\underline{01}\;11001010.$$

However, note from the above, that $\mathcal{E}_2(X) = X^{(46)}\,C_4$ can also be chosen because

$$h_{bal}(X) = \{3, 4\} \subseteq [0, 7].$$

With regard to decoding, on receiving the 2-OSN word

$$YC = 10100010110111001010,$$

the sequence of the last $r = 8$ bits (i.e., $C = 11001010$) is the check word that allows to identify the m_1-balancing function $\langle\,\Gamma_h\rangle$ used in the encoding process. Since $C \in \Gamma_3$, the remaining word Y is decoded into the m_0-balanced word

$$\mathcal{E}_2^{-1}(Y\,C) = \langle\,\Gamma_3\rangle^{-1}(Y) = 100000101111.$$

III. Efficient Implementations of the Refined Coding Scheme and the Complexity Analysis

There may be many ways to implement the coding scheme defined in Section II. However, before going into the details, note that given any integer $d \in [0, k(k-1)/2]$ and word $X = x_1 x_2 \ldots x_n \in \mathbb{Z}_2^k$ the word $X^{(d)} \in \mathbb{Z}_2^k$ can be computed in $O(k \log k)$ bit operations with $O(k)$ bit memory elements with a sequence of at most $k/2$ "giant steps" followed by a sequence of at most $2k - 3$ "baby steps"; as done in [20]. In the sequence of "giant steps", for $j = 1, 2, \ldots$, the bit x_j is exchanged with the bit x_{k-j+1} to compute $X^{(i_j)}$ until $i_{j+1} > d$, where

$$i_{j+1} = i_j + 2k - 4j - 3$$

and $i_0 = 0$. As soon as $i_{j+1} > d$, the sequence of "baby steps" follows to compute $X^{(i_j+1)}$, $X^{(i_j+2)}$, …, $X^{(d)}$ from $X^{(i_j)}$. Analogously, given $X^{(d)} \in \mathbb{Z}_2^k$ and $d \in [0, k(k-1)/2]$ the word $X \in \mathbb{Z}_2^k$ can be recovered in $O(k \log k)$ bit operations with $O(k)$ bit memory elements. In this section, we rely on these giant-baby step based algorithms.

Now, a simple way to implement the coding scheme in Section II, relies on computing the entire random walk sequence

$$RWS(X) \overset{\text{def}}{=} \left\{\left(i, X^{(i)}, m_1\left(X^{(i)}\right)\right) : i = 0, 1, \ldots, \frac{k(k-1)}{2}\right\}$$

during the encoding. In this process, whenever $i = d_h$, for $h = 0, 1, \ldots, p-1$, step S1 of Algorithm 1 is executed to check whether $\mu_1 \in m_1(\Gamma_h) = [\alpha_h, \beta_h]$; that is, if there exists an m_1-balancing check $C \in \Gamma_h$. If $\mu_1 \in [\alpha_h, \beta_h]$ then $h_{bal} = h$, $d_{h_{bal}} = i$ and the information part $Y = \left\langle\,\Gamma_{h_{bal}}\right\rangle(X) = X^{(i)}$ is readily available to form a codeword. With regard to the check word, $C_{h_{bal}} \in \Gamma_{h_{bal}}$, to be appended to Y, it can be computed with a table look-up, like the example Table I, indexed by the readily available index h and the readily available m_1-unbalance μ_1 in (24). It is readily seen that such strategy essentially takes time $T = O(k^2 \log k)$ bit operations to compute the entire random walk sequence $RWS(X)$. With regard to the space complexity, if we assume that the partition $\mathcal{CS}$ in Definition 1 is chosen so that (18) holds with equality then

$$S = O\left(\left|\bigcup_{h=0}^{p-1} \Gamma_h\right| \cdot r\right) = O(k^2 r)$$

memory bits are essentially needed to store the check word table. With regard to the decoding, for simplicity, assume that the function $\left(i, X^{(i)}\right) \to X$ is computed sequentially with $T = O(k \log k)$ bit operations and $S = O(k)$ memory bits as the giant-baby step based decoding algorithm in [20]. In this case, on receiving a codeword $YC = X^{(d_{h_{bal}})}C$, a table look-up indexed by C can be maintained to compute $i = d_{h_{bal}}$ from C. Once $d_{h_{bal}}$ is known, X can be computed from $(d_{h_{bal}}, Y)$ with the giant-baby step based algorithm explained above. In this way, $T = O(k \log k)$ bit operations are essentially needed to compute X from $(d_{h_{bal}}, Y)$ and $S = O(k^2 \log k)$ memory bits are essentially needed to compute $d_{h_{bal}} \in [0, k(k-1)/2]$ from $C \in \bigcup_{h=0}^{p-1} \Gamma_h$ with the table look-up. Note that, if the codes are close to optimal then $r = O(\log k)$ because of (3). In this case, this implementation requires $T = O(k^2 \log k)$ bit operations with $S = O(k^2 \log k)$ memory bits for encoding and $T = O(k \log k)$ bit operations with $S = O(k^2 \log k)$ memory bits for decoding. Also, note that with the above complexities all possible encodings of a given m_0-balanced data word can be computed and, as mentioned earlier, this may be useful to convey some more extra information as done in [12], [16], and [29] for balanced codes.

When k is relatively very big, the above implementation may result prohibitively very complex in terms of time and space. In the remaing part of this section, we show another implementation of the coding scheme presented in Section II which computes one codeword for any given m_0-balanced data word and requires $O(k \log k)$ bit operations with $O(k)$ bit memory elements, as claimed in the abstract. Two major coding problems need to be addressed and efficiently solved in the General Encoding Algorithm 1 for a given m_0-balanced data word X:

1) find one balancing index $h_{bal} \in h_{bal}(X)$ and compute the relative information part $Y = \langle \Gamma_{h_{bal}} \rangle(X) = X^{(d_{h_{bal}})}$; and
2) compute the relative check word $C_{h_{bal}} \in \Gamma_{h_{bal}}$ to be appended to Y.

Here, instead of going into details in designing efficient algorithms directly solving the first problem, we efficiently reduce this problem to the coding problem of finding an m_1-crossing index i_{cx} of X defined in Definition 2. In this reduction, from a given i_{cx}, the balancing index h_{bal} is computed easily with Algorithm 3 below as the balancing index, $\hat{h} \in [0, p-1]$, defined in the proof of Theorem 2. Algorithm 3 also efficiently computes $d = d_{h_{bal}}$. The information part Y is then computed from d and X with the above giant-baby step method. With regard to the second problem, by using the enumerative coding technique in [4], the check word $C_{h_{bal}}$ of the codeword is computed as the $(h_{bal}+1)$th element in $S(r, r/2, \mu_1)$, with μ_1 being the m_1-unbalance in (24). All this can be done by maintaining a table look-up of size $O((\log k)^5)$ bit memory elements. In Sub-section III-A we give the encoding algorithm with the complexity analysis and in Sub-section III-B we analyze the associated decoding algorithm. In the following, for simplicity, assume $S(r, r/2) = \bigcup_{h=0}^{p-1} |\Gamma_h|$.

A. Encoding

Refer to Table I for an example. Let

$$\hat{h} : \left[0, \frac{k(k-1)}{2} + \epsilon(k,r)\right) \cap \mathbb{N} \to [0, p-1] \quad (25)$$

be the integer value function defined as

$$\hat{h}(i) \overset{\text{def}}{=} \begin{cases} \hat{h} \text{ if } i \in I_{\hat{h}} \text{ and } \hat{h} \in [1, p-1] \\ 0 \text{ if } i \in I_0 \cup I_p; \end{cases}$$

where $\epsilon(k,r)$ is defined in (18) and the integer intervals I_0, I_1, …, I_p are defined in (11). Note that, if (18) holds then $\hat{h}(i)$ is a well defined integer function such that

1) there exists an m_1-crossing index, $i_{cx} \in [0, k(k-1)/2 + \epsilon(k,r))$ according to Definition 2;

and

2) if $i_{cx} \in [0, k(k-1)/2 + \epsilon(k,r))$ is an m_1-crossing index then $\hat{h}(i_{cx}) \in [0, p-1]$ is a balancing index with respect to the refined scheme. In fact, $\hat{h}(i_{cx})$ is exactly equal to the integer, $\hat{h}$, defined in the proof of Theorem 2. (26)

For example, in Table I, $\hat{h}(0\text{–}8) = 0$, $\hat{h}(9\text{–}21) = 1$, …, $\hat{h}(61) = 7$ and $\hat{h}(62\text{–}65) = 0$. Also, for the m_0-balanced information word X in Example 5, $i_{cx} = 44$ is an m_1-crossing index and so $\hat{h}(44) = 4$ is a refined scheme balancing index. Now, given any $i \in [0, k(k-1)/2 + \epsilon(k,r))$, the indices $\hat{h}(i)$ and $d_{\hat{h}(i)}$ can be computed efficiently with Algorithm 3 below, whose idea is to compute partial areas of the "bell-shaped histogram" in Table I. In the algorithm, refer to Theorem 3 in the Appendix, Table I and note that

$$\alpha = \min_{C \in S(r,r/2)} m_1(C) = \frac{r(r+2)}{8} \in \mathbb{N},$$
$$\beta = \max_{C \in S(r,r/2)} m_1(C) = \alpha + \frac{r^2}{4} \in \mathbb{N}. \quad (27)$$

Also, for simplicity, assume to have a table look-up of size $O(r^3)$ bits which stores the integer sequence

$$\left\{ s(\mu) \overset{\text{def}}{=} s(r, r/2, \mu) : \mu \in \left[\alpha - 1, \left\lfloor \frac{\alpha+\beta}{2} \right\rfloor\right] \cap \mathbb{N} \right\} \quad (28)$$

of the $O(r^2)$ integer numbers defined in (9), each of which of size r bits. Furthermore, let

$$i_0(\alpha - 1) = -\left\lceil \frac{|\Gamma_0|}{2} \right\rceil \quad (29)$$

and, for all integer $\mu \in [\alpha, \lfloor(\alpha+\beta)/2\rfloor]$, let

$$\begin{aligned} i_0(\mu) &\overset{\text{def}}{=} \sum_{h=0}^{s(\mu)-1} |\Gamma_h| - \left\lceil \frac{|\Gamma_0|}{2} \right\rceil \\ &= \left(\sum_{h=0}^{s(\mu-1)-1} |\Gamma_h| - \left\lceil \frac{|\Gamma_0|}{2} \right\rceil \right) + \sum_{h=s(\mu-1)}^{s(\mu)-1} |\Gamma_h| \\ &= i_0(\mu-1) + \sum_{h=s(\mu-1)}^{s(\mu)-1} |\Gamma_h|. \end{aligned} \tag{30}$$

Note that, 1) the quantity "$i_0(\mu) + \lceil|\Gamma_0|/2\rceil = \sum_{h=0}^{s(\mu)-1} |\Gamma_h|$ is equal to the area of the bell-shaped histogram in Table I until the $(\mu-\alpha+1)$th lower corner (in the order from left to right)"; 2) the integer interval

$$\begin{aligned} I_{s(\mu)} &= \left[d_{s(\mu)} - \left\lceil \frac{|\Gamma_{s(\mu)}|}{2} \right\rceil + 1, d_{s(\mu)} + \left\lfloor \frac{|\Gamma_{s(\mu)}|}{2} \right\rfloor \right] \\ &= [i_0(\mu)+1, i_0(\mu) + |\Gamma_{s(\mu)}|] \end{aligned} \tag{31}$$

because of (10) and (11); and, 3) if $h \in [s(\mu-1), s(\mu))$ then the quantity $|\Gamma_h|$ is constant and

$$|\Gamma_h| = |\Gamma_0| - 2(\mu-\alpha) = (\beta-\alpha+1) - 2(\mu-\alpha). \tag{32}$$

So, (29), (30) and (32) give,

$$i_0(\mu) = \begin{cases} -\lceil|\Gamma_0|/2\rceil & \text{if } \mu = \alpha-1, \\ i_0(\mu-1) + [s(\mu)-s(\mu-1)][|\Gamma_0| - 2(\mu-\alpha)] \\ \qquad\qquad \text{if } \mu \in [\alpha, \lfloor(\alpha+\beta)/2\rfloor]. \end{cases} \tag{33}$$

```
Algorithm 3 to compute $\hat{h}(i)$ and relative $d_{\hat{h}}$ from $i$
Input: the integer $i \in [0, k(k-1)/2 + \epsilon(k,r))$.
Output: the integer $\hat{h}(i) \in [0, p-1]$ defined in (25) and $d_{\hat{h}} \in \{d_h : h = 0,1,\ldots,p-1\}$, where the integers $d_h$ are defined in (10).
Perform steps S0, S1, S2 and S3.
S0: if
    $i \in \left[0, \left\lfloor \frac{|\Gamma_0|}{2} \right\rfloor\right] \cup \left[\left\lfloor \frac{|\Gamma_0|}{2} \right\rfloor + \sum_{h=1}^{p-1} |\Gamma_h| + 1, \frac{k(k-1)}{2} + \epsilon(k,r)\right)$
    then let $\hat{h} = 0$ and $d_{\hat{h}} = 0$ according to the definition (25) of $\hat{h}$. Hence, go to step S3.
S1: compute the largest integer $\hat{\mu} \in [\alpha, \lfloor(\alpha+\beta)/2\rfloor]$ such that $i_0(\hat{\mu}) < i$, where the function $i_0(\mu)$ is given in (33). Also, compute $i_0(\hat{\mu})$. Note that for such $\hat{\mu}$ we have $i_0(\hat{\mu}) < i \le i_0(\hat{\mu}+1)$.
S2: according to (31) and (32), execute the steps S2.1, S2.2, S2.3 and S2.4.
    S2.1: compute $\gamma = |\Gamma_{\hat{h}}|$ as follows.
        $\gamma = |\Gamma_0| - 2[(\hat{\mu}+1) - \alpha]$.
    S2.2: compute the quotient
        $\phi = \left\lfloor \frac{i - i_0(\hat{\mu}) - 1}{\gamma} \right\rfloor$.
    S2.3: compute
        $\hat{h} = s(\hat{\mu}) + \phi$.
    S2.4: compute
        $d_{\hat{h}} = i_0(\hat{\mu}) + \phi \cdot \gamma + \left\lceil \frac{\gamma}{2} \right\rceil$
S3: output $\hat{h}$, $d_{\hat{h}}$ and exit.
```

Example 6 [For $k = 12$, $r = 8$ and $p = 8$ (Continued)]: In the example given in Table I, if $i = 44 \in [0, 65]$ is given as input to Algorithm 3 then it executes as follows.

S0: *since $i = 44 \notin [0, 8] \cup [62, 65]$ continue to step S1.*

S1: *compute the largest integer $\hat{\mu} \in [\alpha, \lfloor(\alpha+\beta)/2\rfloor] = [10, 18]$ such that $i_0(\hat{\mu}) < i$. From (33), the encoder computes*

$$\begin{aligned} i_0(\alpha) &= -9 + (1-0)(17) = 8 < 44, \\ i_0(\alpha+1) &= 8 + (1-1)(15) = 8 < 44, \\ i_0(\alpha+2) &= 8 + (2-1)(13) = 21 < 44, \\ i_0(\alpha+3) &= 21 + (3-2)(11) = 32 < 44, \\ i_0(\alpha+4) &= 32 + (5-3)(9) = 50 \ge 44. \end{aligned}$$

So, $\hat{\mu} = \alpha + 3 = 13$.

S2: *execute the steps S2.1, S2.2, S2.3 and S2.4.*

S2.1: *compute $\gamma = 17 - 2(14 - 10) = 9$.*

S2.2: *compute*

$$\phi = \left\lfloor \frac{44 - 32 - 1}{9} \right\rfloor = \left\lfloor \frac{11}{9} \right\rfloor = 1.$$

S2.3: *compute $\hat{h} = s(13) + 1 = 4$.*

S2.4: *compute $d_{\hat{h}} = 32 + 1 \cdot 9 + \lceil 9/2 \rceil = 46$.*

S3: *output $\hat{h} = 4$, $d_{\hat{h}} = 46$ and exit.*

With regard to the space complexity of Algorithm 3, the predominant term is the size of the table look-up storing the integer sequence (28). Hence, Algorithm 3 requires a size of $O(r^3)$ bits. With regard to the time complexity, note that step S0 can be accomplished in time $O(r)$ bit operations because the size of $i = O(k^2)$ is $O(\log i) = O(\log k) = O(r)$; S1 can be accomplished in time $O(r^3 \log r)$ bit operations because, with (33), S1 can be computed with $O(\beta - \alpha) = O(r^2)$ substeps (see (27)), each of which performing $O(\log|\Gamma_0|) = O(\log(\beta-\alpha)) = O(\log r)$ additions (see (32)) of numbers of size $O(\log i) = O(r)$ bits; S2 can be accomplished in time $O(r^2)$ because step S2.1 takes time $O(\log r)$ bit operations, S2.2 takes time $O(\log^2 k) = O(r^2)$ bit operations, S2.3 takes time $O(r)$ bit operations, S2.4 takes time $O(\log^2 k) = O(r^2)$ bit operations; and, finally, step S3 takes time $O(1)$ bit operations. Hence, Algorithm 3 requires a time complexity of $O(r^3 \log r)$ bit operations.

At this point, using Algorithm 3 and the enumerative coding technique in [4] the efficient encoding Algorithm 4 can be given below. So, as in Table I, let us assume that the check words in each $S(r, r/2, \mu)$ are arranged in some lexicographic order. In Algorithm 4, once a balancing index h_{bal} is chosen, say, by using property (26), as $h_{bal} = \hat{h}(i_{cx})$, the check word is computed in step S1.5 as the $(h_{bal} + 1)$th element in $S(r, r/2, \mu_1)$, with μ_1 being the quantity which makes the overall codeword m_1-balanced. To this aim, for simplicity, assume to have a table look-up of size $O(r^5)$ bits which stores the set

$$\{s(a, b, c) : a, b \in [0, r] \text{ and } c \in [0, r(r + 1)/2]\} \quad (34)$$

of $O(r^4)$ integer numbers in (9) each of which has size r bits.

Algorithm 4 Efficient Encoding Algorithm

Input: the information word $Z \in \mathbb{Z}_2^{\tilde{k}}$.
Output: the codeword

$$YC = \mathcal{E}_2(\mathcal{E}_1(Z)) = (\mathcal{E}_1 \circ \mathcal{E}_2)(Z) \in \mathcal{SN}(n, 2)$$

where $n = k + r$, $Y \in \mathbb{Z}_2^k$ and $C \in \mathbb{Z}_2^r$.
Perform steps S0, S1, S2 and S3.
S0 apply any m_0-balancing method to Z and let X be the m_0-balanced word of length k associated with Z, i.e. $X = \mathcal{E}_1(Z)$. In this way, $m_0(X) = k/2$.
S1 (find the balancing index of X and compute the check word): execute the steps S1.1–S1.5.
S1.1: find an m_1-crossing index of X according to Definition 2.
S1.2: compute $h_{bal} = \hat{h}(i_{cx})$ and the relative $d_{h_{bal}}$ with Algorithm 3.
S1.3: compute $\langle \Gamma_{h_{bal}} \rangle(X) = X^{(d_{h_{bal}})}$ and $m_1(\langle \Gamma_{h_{bal}} \rangle(X))$.
S1.4: compute

$$\mu_1 = \frac{n(n+1)}{4} - m_1\left(\langle \Gamma_{h_{bal}} \rangle(X)\right) - \frac{kr}{2}.$$

S1.5: With the enumerative coding technique, compute the $(h_{bal} + 1)$th element in $S(r, r/2, \mu_1)$. Let it be C_{h_{bal},μ_1}.
S2: Since h_{bal} is a balancing index of X, select the m_1-balanced encoding word of X as

$$\mathcal{E}_2(X) = YC = \langle \Gamma_{h_{bal}} \rangle(X) C_{h_{bal},\mu_1} = X^{(d_{h_{bal}})} C_{h_{bal},\mu_1}$$

In this way the data word Z is encoded as $\mathcal{E}_2(X) = YC = \mathcal{E}_2(\mathcal{E}_1(Z))$.
S3: output $\mathcal{E}_2(X) = X^{(d_{h_{bal}})} C_{h_{bal},\mu_1}$ and **exit**.

Example 7 [For $k = 12$, $r = 8$ and $p = 8$ (Continued)]: As in Example 5, suppose $X =$ 100000101111 *is the given m_0-balanced word which needs to be encoded into a word of $\mathcal{SN}(20, 2)$. In this case, Algorithm 4 executes as follows.*

S0 *by applying the choosen m_0-balancing method to Z, the output is $X =$* 100000101111.

S1 *(find the balancing index of X and compute the check word): execute the steps S1.1–S1.5.*

S1.1*: find an m_1-crossing index of X. From the random walk in Example 5, $i_{cx} = 44 \in [0, 65]$ is such that $m_1(X^{(i_{cx})}) = k(k + 1)/4 = 39$. Note that, for the chosen example word X there is only one m_1-crossing index. But, in general, there may be many such indices.*

S1.2*: compute $h_{bal} = \hat{h}(i_{cx})$ and the relative $d_{h_{bal}}$ with Algorithm 3. From Example 6,*

$$h_{bal} = \hat{h}(44) = 4$$

and

$$d_{h_{bal}} = d_4 = 46.$$

S1.3*: compute*

$$\langle \Gamma_4 \rangle(X) = X^{(46)} = \underline{11}00010\underline{11\underline{001}}$$

and

$$m_1(\langle \Gamma_4 \rangle(X)) = 38.$$

S1.4*: compute*

$$\mu_1 = \frac{n(n+1)}{4} - m_1\left(\langle \Gamma_4 \rangle(X)\right) - \frac{kr}{2} = 105 - 38 - 48 = 19.$$

S1.5*: With the enumerative coding technique, compute the $(h_{bal}+1)$th = 5th element in $S(8, 4, 19)$. It is (see Table I)*

$$C_{4,19} = 01100101.$$

S2*: Compute*

$$\begin{aligned}\mathcal{E}_2(X) = Y\,C &= X^{(46)}\, C_{4,19}\\ &= \underline{11}00010\underline{11\underline{001}}\,01100101 \in \mathcal{SN}(20, 2).\end{aligned}$$

S3*: Output $\mathcal{E}_2(X) =$ 110001011001 01100101 and exit.*

With regard to the complexity of Algorithm 4, note that step S0 can be accomplished in space $O(k)$ memory bits and time $O(k \log k)$ bit operations by using any of the methods given in [1], [2], [7], [11], [12], [16], [18], [19], [21], and [29]. S1 can be accomplished in space $O(r^5 + k)$ memory bits and time $O(r^3)$ bit operations because step S1.1 takes $O(k)$ memory bits and $O(k \log k)$ bit operations if done with the giant-baby step method explained at the beginning of this section, S1.2 takes $O(r^3)$ memory bits and $O(r^3 \log r)$ bit operations, S1.3 takes $O(k)$ memory bits and $O(k \log k)$ bit operations if done with the giant-baby step method, S1.4 takes $O(\log k)$ memory bits and $O(\log k)$ bit operations, S1.5 takes $O(r)$ additions of r-bit numbers so, with the table look-up (34) of size $O(r^5)$ memory bits, it takes time $O(r^2)$ bit operations; and, finally, step S2 takes time $O(1)$ bit operations. Hence, Algorithm 4 has a space complexity of $O(r^5 + k) = O(k)$ memory bits and a time complexity of $O(r^3 \log r + k \log k) = O(k \log k)$ bit operations because of (3).

B. Decoding

In Algorithm 4 the check word C encodes the balancing index h_{bal} through the index of C in the lexicographically ordered set $S(r, r/2, m_1(C))$. Hence, during the decoding process, once this lexicogrphic index h_{bal} is computed, an efficient algorithm is needed which computes $d_{h_{bal}}$ from h_{bal}. Once $d_{h_{bal}}$ is known, X can be recovered easily from $d_{h_{bal}}$ and Y with the giant-baby step method of [20]. The algorithm which computes $d_{\hat{h}}$ from any $\hat{h}$ is given below. As in Algorithm 3, it is assumed to have a table look-up storing the integer sequence in (28).

Algorithm 5 to compute $d_{\hat{h}}$ from $\hat{h}$

Input: the integer $\hat{h} \in [0, p-1]$.
Output: the integer $d_{\hat{h}} \in \{d_h : h = 0, 1, \ldots, p-1\}$, where the d_h are defined in (10).
Perform steps S1, S2 and S3.
S1: compute the unique integer $\hat{\mu} \in [\alpha, \lfloor(\alpha+\beta)/2\rfloor]$ such that $s(\hat{\mu}-1) \leq \hat{h} < s(\hat{\mu})$. Also, compute $i_0(\hat{\mu}-1)$, where the function $i_0(\mu)$ is given in (33).
S2: execute the steps S2.1 and S2.2.
S2.1: compute $\gamma = |\Gamma_{\hat{h}}| = |\Gamma_0| - 2(\hat{\mu}-\alpha)$.
S2.2: compute

$$d_{\hat{h}} = i_0(\hat{\mu}-1) + [\hat{h} - s(\hat{\mu}-1)]\gamma + \left\lceil \frac{\gamma}{2} \right\rceil.$$

S3: output $d_{\hat{h}}$ and **exit**.

Example 8 [For $k = 12$, $r = 8$ and $p = 8$ (Continued)]: In the example given in Table I, if $\hat{h} = 4 \in [0, 7]$ is given as input to Algorithm 5 then it executes as follows.

S1*: compute the unique integer $\hat{\mu} \in [\alpha, \lfloor(\alpha+\beta)/2\rfloor] = [10, 18]$ such that $s(\hat{\mu}-1) \leq \hat{h} < s(\hat{\mu})$. Also, compute $i_0(\hat{\mu}-1)$. From Table I and (33), the decoder computes*

$$\begin{aligned}
&\hat{h} = 4 \notin [s(9), s(10)) = [0, 1) \;\; \textit{and} \\
&\qquad\qquad i_0(\alpha-1) = -9, \\
&\hat{h} = 4 \notin [s(10), s(11)) = [1, 1) \;\; \textit{and} \\
&\qquad\qquad i_0(\alpha) = -9 + (1-0)(17) = 8, \\
&\hat{h} = 4 \notin [s(11), s(12)) = [1, 2) \;\; \textit{and} \\
&\qquad\qquad i_0(\alpha+1) = 8 + (1-1)(15) = 8, \\
&\hat{h} = 4 \notin [s(12), s(13)) = [2, 3) \;\; \textit{and} \\
&\qquad\qquad i_0(\alpha+2) = 8 + (2-1)(13) = 21, \\
&\hat{h} = 4 \in [s(13), s(14)) = [3, 5) \;\; \textit{and} \\
&\qquad\qquad i_0(\alpha+3) = 21 + (3-2)(11) = 32.
\end{aligned}$$

So, $\hat{\mu} = \alpha + 4 = 14$.

S2*: execute the steps S2.1 and S2.2.*

S2.1*: compute*

$$\gamma = |\Gamma_{\hat{h}}| = |\Gamma_0| - 2(\hat{\mu}-\alpha) = 17 - 2(14-10) = 9.$$

S2.2*: compute*

$$\begin{aligned}
d_{\hat{h}} &= i_0(\hat{\mu}-1) + [\hat{h} - s(\hat{\mu}-1)]\gamma + \left\lceil \frac{\gamma}{2} \right\rceil \\
&= 32 + (4-3)\cdot 9 + 5 = 46.
\end{aligned}$$

S3*: output $d_{\hat{h}} = 46$ and exit.*

As the above complexity analysis of Algorithm 3, we conclude that Algorithm 5 can execute with $O(r^3)$ memory bits and $O(r^3 \log r)$ bit operations.

At this point, the decoding algorithm can be given below. As Algorithm 4, assume to have a table look-up of size $O(r^5)$ bits storing the set (34) of integers.

Algorithm 6 Efficient Decoding Algorithm

Input: the codeword associated with some data word $Z \in \mathbb{Z}_2^{\tilde{k}}$,

$$YC = \mathcal{E}_2(\mathcal{E}_1(Z)) = (\mathcal{E}_1 \circ \mathcal{E}_2)(Z) \in \mathcal{SN}(n, 2).$$

with $n = k + r$, $Y \in \mathbb{Z}_2^k$ and $C \in \mathbb{Z}_2^r$.
Output: the information word $Z = (\mathcal{E}_1 \circ \mathcal{E}_2)^{-1}(YC)$.
Perform steps S0, S1, S2 and S3.
S0: compute the index $h_{bal} \in [0, p-1]$ such that $C \in \Gamma_{h_{bal}}$.
Perform steps S0.1, S0.2 and S0.3.
S0.1: compute $\mu_1 = m_1(C)$.
S0.2: compute the lexicographic index, say

$$\xi_{\mu_1}(C) \in [0, s(r, r/2, \mu_1)),$$

of C in $S(r, r/2, \mu_1)$.
S0.3: set

$$h_{bal} = \xi_{\mu_1}(C) \in [0, s(r, r/2, \mu_1)) \subseteq [0, p-1].$$

S1: compute the word

$$X = \mathcal{E}_2^{-1}(YC) = \langle \Gamma_{h_{bal}} \rangle^{-1}(Y).$$

Perform steps S1.1, S1.2.
S1.1: set $\hat{h} = h_{bal} \in [0, p-1]$ and run Algorithm 5 to compute

$$d_{\hat{h}} = d_{h_{bal}} \in \left[0, \frac{k(k-1)}{2} + \epsilon(k, r)\right)$$

from $\hat{h}$.
S1.2: from $d_{h_{bal}}$, compute the word $X = \langle \Gamma_{h_{bal}} \rangle^{-1}(Y)$ with the giant-baby step method in [20].
S2: undo the m_0-balancing encoding chosen in step S0 of Algorithm 4 on the word X. Let $Z = \mathcal{E}_1^{-1}(X) \in \mathbb{Z}_2^{\tilde{k}}$ be the information word whose m_0-balanced encoding is X.
S3: output the word

$$Z = \mathcal{E}_1^{-1}(X) = \mathcal{E}_1^{-1}(\mathcal{E}_2^{-1}(YC)) = (\mathcal{E}_1 \circ \mathcal{E}_2)^{-1}(YC)$$

and **exit**.

Example 9 [For $k = 12$, $r = 8$ and $p = 8$ (Continued)]: Suppose

$$\begin{aligned}
\mathcal{E}_2(X) = Y\,C &= X^{(46)}\,C_{4,19} \\
&= \underline{11}0001\underline{11001}\,01100101 \in \mathcal{SN}(20, 2)
\end{aligned}$$

is given as input to decoding Algorithm 6. It executes as follows.

S0*: compute the index $h_{bal} \in [0, p-1]$ such that*

$$C = 01100101 \in \Gamma_{h_{bal}}.$$

Perform steps S0.1, S0.2 and S0.3.

S0.1*: compute $\mu_1 = m_1(01100101) = 19$.*

S0.2*: compute the lexicographic index (see Table I)*

$$\xi_{19}(01100101) = 4 \in [0, s(8, 4, 19)).$$

S0.3*: set $h_{bal} = \xi_{19}(C) = 4 \in [0, 7]$.*

S1*: compute the word $X = \mathcal{E}_2^{-1}(YC) = \langle \Gamma_{h_{bal}} \rangle^{-1}(Y)$. Perform steps S1.1, S1.2.*

S1.1*: compute*

$$d_{\hat{h}} = d_{h_{bal}} = 46 \in [0, 66)$$

from $\hat{h} = h_{bal} = 4 \in [0, 7]$ with Algorithm 5.

S1.2: *from $d_{h_{bal}} = 46$, compute the word*

$$X = \mathcal{E}_2^{-1}(YC) = \langle \Gamma_4 \rangle^{-1}(X^{(46)}) = 100000101111$$

as in [20].

S2: *undo the m_0-balancing encoding chosen in step S0 of Algorithm 4 on the word X.*

S3: *output the word $Z = \mathcal{E}_1^{-1}(X)$ and exit.*

Again, this can be done with a size of $O(r^5 + k) = O(k)$ memory bits and $O(r^3 \log r + k \log k) = O(k \log k)$ bit operations because of (3).

IV. Concluding Remarks

A new efficient method to design second-order spectral-null codes is given. These new codes are obtained by applying the refined parallel decoding scheme method for the balanced code designs in [1], [12], and [19] to the random walk method for second-order spectral-null code design in [20]. This gives new non-recursive efficient code designs which make a good use of non-second-order spectral-null check words. For this reason the proposed codes are considerably less redundant than the code designs found in the literature (which are all recursive). In particular, if $k \in 2\mathbb{N}$ is the length of a 1-OSN code then the new 2-OSN coding scheme has length $n = k + r \in 4\mathbb{N}$ with an extra redundancy, $r \in 2\mathbb{N}$ such that $r \simeq 2\log_2 k + (1/2)\log_2 \log_2 k - 0.174$ check bits. Table II compares the proposed non-recursive codes with the optimal codes and also with the recursive codes given in [20]. It is also shown how these codes can be implemented with $O(k \log k)$ bit operations and $O(k)$ bit memory elements. If $k \notin 2\mathbb{N}$ or $r \notin 2\mathbb{N}$ then essentially the same design can be given to encode information into the set of words $S(n, \mu_0, \mu_1)$ for some fixed $\mu_0, \mu_1 \in \mathbb{N}$. In particular, if $k, r \in 2\mathbb{N} + 1$ and $n \in 4\mathbb{N}$ then efficient designs can be obtained to convert the information words to words in the set $S(n, n/2, n^2/2)$ (or, $S(n, n/2, n(n+2)/2)$). Note that this non-recursive coding scheme is very general and it can be implemented in many different ways. These may depend on k, r, p, the choice of the particular "m_1-balancing random walk" and the choice of the partition $\mathcal{CS}$. Also, note that the decoding can be done in parallel because the check word directly identifies the m_1-balancing function used to encode and each balancing function with its inverse is very simple to compute because they are the composition of a partial reversion of a k-bit word and a cyclic shift. In general, parallel algorithms can be applied for both encoding and decoding as done in [19].

Appendix

Theorem 3: Given $r \in \mathbb{N}$, let

$$s(r, \mu_0, \mu_1) \overset{\text{def}}{=} |S(r, \mu_0, \mu_1)|, \quad \textit{for all integer } \mu_0, \mu_1 \in \mathbb{N}.$$

The following statements hold.

1) *Given $\mu_0 \in [0, r]$, if $C \in S(r, \mu_0)$ then the minimum possible value for $m_1(C)$ is*

$$\alpha \overset{\text{def}}{=} \min_{C \in S(r,\mu_0)} m_1(C) = \frac{\mu_0(\mu_0 + 1)}{2} \in \mathbb{N}.$$

2) *Given $\mu_0 \in [0, r]$, if $C \in S(r, \mu_0)$ then the maximum possible value for $m_1(C)$ is*

$$\beta \overset{\text{def}}{=} \max_{C \in S(r,\mu_0)} m_1(C) = \alpha + \mu_0(r - \mu_0) \in \mathbb{N}.$$

3) *The integer sequence*

$$\{s(r, \mu_0, \mu) : r, \mu_0, \mu_1 \in \mathbb{N}\}$$

satisfies the following recurrence relation.

$$s(r, \mu_0, \mu_1) = s(r-1, \mu_0, \mu_1) + s(r-1, \mu_0 - 1, \mu_1 - r),$$

with initial conditions

$$s(r, \mu_0, \mu_1) = \begin{cases} 0 & \textit{if } \mu_1 < \alpha \textit{ or } \mu_1 > \beta, \\ 1 & \textit{if } \mu_1 = \alpha \textit{ or } \mu_1 = \beta; \end{cases}$$

4) *Given $\mu_0 \in [0, r]$, the integer sequence*

$$\{s(r, \mu_0, \mu) : \mu = \alpha, \alpha + 1, \ldots, \beta\}$$

is unimodal and symmetric with respect to

$$\mu_{mean} \overset{\text{def}}{=} \frac{\alpha + \beta}{2} = \frac{\mu_0(r+1)}{2} \in \mathbb{R};$$

namely,

$$\begin{aligned} & s(r, \mu_0, \alpha) \\ & \le s(r, \mu_0, \alpha + 1) \le \ldots\ldots \le s\left(r, \mu_0, \left\lfloor \frac{\mu_0(r+1)}{2} \right\rfloor\right) \\ & = \max_{\mu \in [\alpha, \beta]} |S(r, \mu_0, \mu)| \\ & = s\left(r, \mu_0, \left\lceil \frac{\mu_0(r+1)}{2} \right\rceil\right) \ge \ldots\ldots \\ & \ge s(r, \mu_0, \beta - 1) \ge s(r, \mu_0, \beta) \end{aligned}$$

and

$$s(r, \mu_0, \alpha + \mu) = s(r, \mu_0, \beta - \mu),$$

for all integers $\mu \in [0, \mu_0(r - \mu_0)]$.

Proof: Property 1), 2) and 3) follow from [20]. Property 4) follows from [22, Th. 2]. ■

Acknowledgment

The authors thank a lot the anonymous referees for their helpful comments and suggestions.

References

[1] S. Al-Bassam and B. Bose, "On balanced codes," *IEEE Trans. Inf. Theory*, vol. 36, no. 2, pp. 406–408, Mar. 1990.
[2] S. Al-Bassam and B. Bose, "Design of efficient balanced codes," *IEEE Trans. Comput.*, vol. 43, no. 3, pp. 362–365, Mar. 1994.
[3] B. Bose, "On unordered codes," *IEEE Trans. Comput.*, vol. 40, no. 2, pp. 125–131, Feb. 1991.
[4] T. M. Cover, "Enumerative source encoding," *IEEE Trans. Inf. Theory*, vol. 19, no. 1, pp. 73–77, Jan. 1973.
[5] K. A. S. Immink, "Spectrum shaping with DC^2-constrained channel codes," *Philips J. Res.*, vol. 40, no. 1, pp. 40–53, 1985.
[6] K. A. S. Immink, *Codes for Mass Data Storage Systems*, 2nd ed. Eindhoven, The Netherlands: Shannon Foundation Publishers, 2004.
[7] D. E. Knuth, "Efficient balanced codes," *IEEE Trans. Inf. Theory*, vol. 32, no. 1, pp. 51–53, Jan. 1986.

[8] R. Mascella, D. Pelusi, L. Pezza, S. Elmougy, L. G. Tallini, and B. Bose, "On efficient second-order spectral-null codes using sets of m_1-balancing functions," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, Jul. 2013, pp. 141–145.

[9] R. Mascella and L. G. Tallini, "On efficient high-order spectral-null codes over the m-ary alphabet," *J. Discrete Math. Sci. Cryptogr.*, vol. 8, no. 3, pp. 459–481, 2005.

[10] R. Mascella and L. G. Tallini, "Efficient m-ary balanced codes which are invariant under symbol permutation," *IEEE Trans. Comput.*, vol. 55, no. 8, pp. 929–946, Aug. 2006.

[11] D. Pelusi, L. G. Tallini, and B. Bose, "On m-ary balanced codes with parallel decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Austin, TX, USA, Jun. 2010, pp. 1305–1309.

[12] D. Pelusi, S. Elmougy, L. G. Tallini, and B. Bose, "m-ary balanced codes with parallel decoding," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3251–3264, Jun. 2015.

[13] L. Pezza, L. G. Tallini, and B. Bose, "Variable length unordered codes," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 548–569, Feb. 2012.

[14] R. M. Roth, P. H. Siegel, and A. Vardy, "High-order spectral-null codes—Constructions and bounds," *IEEE Trans. Inf. Theory*, vol. 40, no. 6, pp. 1826–1840, Nov. 1994.

[15] V. Skachek, T. Etzion, and R. M. Roth, "Efficient encoding algorithm for third-order spectral-null codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 846–851, Mar. 1998.

[16] T. G. Swart and J. H. Weber, "Efficient balancing of q-ary sequences with parallel decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Seoul, South Korea, Jun./Jul. 2009, pp. 1564–1568.

[17] L. G. Tallini, S. Al-Bassam, and B. Bose, "Feedback codes achieving the capacity of the Z-channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 1357–1362, Mar. 2008.

[18] L. G. Tallini, R. M. Capocelli, and B. Bose, "Design of some new efficient balanced codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 790–802, May 1996.

[19] L. G. Tallini and B. Bose, "Balanced codes with parallel encoding and decoding," *IEEE Trans. Comput.*, vol. 48, no. 8, pp. 794–814, Aug. 1999.

[20] L. G. Tallini and B. Bose, "On efficient high-order spectral-null codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2594–2601, Nov. 1999.

[21] L. G. Tallini and U. Vaccaro, "Efficient m-ary balanced codes," *Discrete Appl. Math.*, vol. 92, no. 1, pp. 17–56, Mar. 1999.

[22] L. G. Tallini, "Geometric properties of high-order spectral-null codes," *Italian J. Pure Appl. Math.*, no. 14, pp. 149–176, 2003.

[23] L. G. Tallini, "Bounds on the capacity of the unidirectional channels," *IEEE Trans. Comput.*, vol. 54, no. 2, pp. 232–235, Feb. 2005.

[24] L. G. Tallini, S. Elmougy, and B. Bose, "Analysis of plain and diversity combining hybrid ARQ protocols over the m($\geq$2)-ary asymmetric channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5550–5558, Dec. 2006.

[25] L. G. Tallini and B. Bose, "On L_1 metric asymmetric/unidirectional error control codes, constrained weight codes and σ-codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, Jul. 2013, pp. 694–698.

[26] C. N. Yang, "Efficient encoding algorithm for second-order spectral-null codes using cyclic bit shift," *IEEE Trans. Comput.*, vol. 57, no. 7, pp. 876–888, Jul. 2008.

[27] C. N. Yang, Z. Y. Lin, and S. L. Peng, "Reducing code length of second-order spectral-null code," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 492–503, Feb. 2015.

[28] J.-H. Youn and B. Bose, "Efficient encoding and decoding schemes for balanced codes," *IEEE Trans. Comput.*, vol. 52, no. 9, pp. 1229–1232, Sep. 2003.

[29] J. H. Weber and K. A. S. Immink, "Knuth's balanced codes revisited," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1673–1679, Apr. 2010.

Luca G. Tallini was born in Frascati (Rome), Italy. He received the Laurea in Mathematics *magna cum laude* from the University of Rome "La Sapienza" in 1991, the M. S. and Ph. D. degrees in Computer Science from Oregon State University, Corvallis OR, in 1994 and 1996 respectively. From June 1997 to June 1998 he had been with the Dipartimento di Informatica ed Applicazioni, University of Salerno, Italy. From June 1998 to December 2002, he had been Assistant Professor at Politecnico di Milano, Italy. Since 2003 he has been with the University of Teramo, Italy, where he is a Professor at the Faculty of Communication Science and an Accademic Senator for the scientific research. His research interests include coding theory, combinatorics, combinatorial algorithms, combinatorial geometry, fault-tolerant computing, parallel computing, neural networks, and VLSI.

Danilo Pelusi received the Ph. D. degree in Computational Astrophysics from the University of Teramo, Teramo, Italy, in 2006. He is with the Faculty of Communication Sciences, University of Teramo, Teramo, Italy. His current research interests include information theory, fuzzy logic and genetic algorithms.

Raffaele Mascella received the Laurea degree in mathematics from the University of L'Aquila in 1995 and the PhD degree from University of Teramo in 2004. From March 2005 to December 2012, he was an assistant professor; since December 2012, he has been an associate professor and the vice-dean of the Faculty of Communication Sciences at the University of Teramo, Italy. His research interests include coding theory, combinatorics, fuzzy logic, neural networks, philosophy of computing and philosophy of mathematics.

Laura Pezza was born in Frosinone, Italy. She received the Laurea degree in Mathematics *magna cum laude* from the University of Rome "La Sapienza" in 1991 and the Dottorato di Ricerca (Ph. D.) degree in Applied Mathematics from the University of Florence in 1996. From September 1996 she has been with the Dipartimento di Scienze di Base ed Applicate per l'Ingegneria, University of Rome "La Sapienza", where she is an Assistant Professor of Numerical Analysis. His research interests include applied mathematics, fluid dynamic, wavelets and refinable functions, signal analysis, and coding theory.

Samir Elmougy received the Ph. D degree in computer science (2005) from the School of Electrical Engineering and Computer Science, Oregon State University, USA; the M. Sc. degree in computer science (1996) and the B. Sc. degree in statistics and scientific computing (1993) both from Mansoura University, Egypt. He is currently an associative professor and the chair of Computer Science Department, Faculty of Computers and Information, Mansoura University, Egypt. From 2008 to 2014, he had been with King Saudi University, Riyadh, Saudi Arabia as an assistant professor of Computer Science at the College of Computers and Information Sciences. His current research interests include algorithms, error correcting codes, computer networks, software engineering and machine learning.

Bella Bose (S'78–M'80–SM'94–F'95) received the B. E. degree in electrical engineering from Madras university, India in 1973; the M. E. degree in electrical engineering from the Indian Institute of Science, Bangalore, in 1975; and the M. S. and Ph. D. degrees in computer science and engineering from Southern Methodist University, Dallas, Texas, in 1979 and 1980, respectively. Since 1980, he has been with Oregon State University where he is a Professor and the Senior Associate Head of the School of Electrical Engineering and Computer Science. His current research interests include error control codes, fault-tolerant computing, parallel processing, and computer networks. He is a fellow of both the ACM and the IEEE.