


Improving the redundancy of Knuth's balancing scheme for packet transmission systems

Elie NGOMSEU MAMBOU*, Ebenezer ESENOGHO, Hendrik FERREIRA

Department of Electrical and Electronic Engineering, University of Johannesburg, Auckland Park, South Africa

Received: 16..10.2018

Accepted/Published Online: 08.04.2019

Final Version: 26.07.2019

Abstract: A simple scheme was proposed by Knuth to generate binary balanced codewords from any information word. However, this method is limited in the sense that its redundancy is twice that of the full sets of balanced codes. The gap between Knuth's algorithm's redundancy and that of the full sets of balanced codes is significantly considerable. This paper attempts to reduce that gap. Furthermore, many constructions assume that a full balancing can be performed without showing the steps. A full balancing refers to the overall balancing of the encoded information together with the prefix. We propose an efficient way to perform a full balancing scheme that does not make use of lookup tables or enumerative coding.

Key words: Balanced codes, binary word, parallel decoding, prefix coding, full balancing

1. Introduction

Balanced codes have been widely studied over the years because of their applicability in the field of communication and in storage structures such as optical and magnetic recording devices like Blu-Ray, DVDs, and CDs [1, 2]; error correction and detection [3, 4]; cable transmission [5]; and noise attenuation in VLSI integrated circuits [6]. For some balancing techniques, the decoding of balanced codes is fast and can be done in parallel, which avoids latency in communication.

A binary word of length k , with k even, is said to be balanced if the number of zeros and ones equals $k/2$. Knuth proposed a simple and efficient scheme to generate balanced codewords [7]. This approach stipulates that any binary word \mathbf{x} of length k can always be encoded into a balanced one denoted as \mathbf{y} by inverting the first e bits of \mathbf{x} , where $1 \leq e \leq k$. The index e is encoded as the prefix \mathbf{p} , which is appended to \mathbf{y} and sent through a channel. At the receiver, the decoder receives the concatenated codeword $\mathbf{y}\mathbf{p}$ and retrieves the original information word through the prefix by inverting back the first e bits of \mathbf{y} . This algorithm is very suitable for long sequences as it does not make use of any lookup tables, neither at the encoder nor at the decoder. A detailed explanation of this method is covered in Section 2.

Many works have been done to reduce the redundancy generated by Knuth's algorithm (KA). In [8], two attempts were described by Weber and Immink; the first one was using the distribution of the prefix index. This consists of setting the encoder to choose smaller values for the prefix index knowing that the position index e might not be unique. By default, KA makes use of the first balancing index while inverting from the least position bits. It has been shown that the distribution of that index for equiprobable information words is not uniform and presents a redundancy that is slightly less than that of KA. The second attempt used the

*Correspondence: emambou@uj.ac.za

multiplicity of balancing points within a word to transmit auxiliary data. The previous schemes provide a fixed length (FL) and variable length (VL) prefix implementation. However, these methods only made a minor improvement on KA.

The second attempt from [8] was exploited in [9] and renamed as bit recycling for Knuth's algorithm (BRKA); it relies on a high probability of having more than one balancing index while performing KA. In other words, this scheme uses the multiplicity of balancing indexes to encode a shorter prefix than that from KA. In [10], a technique for balancing words was presented based on permutations, the arcade game Pac-Man, and limited-precision integers; the redundancy of KA was improved and the redundancy of the full set of balanced codes was nearly achieved at the cost of high complexity and large memory usage.

In [11], a systematic variable-to-fixed length technique was presented for encoding binary source sequences into binary balanced codewords. This method is simple in the sense that its encoding only has to keep track of the sequence's weight while the decoding is done in one step. However, this scheme's redundancy is larger than that of the fixed-to-fixed length schemes for long codes and smaller for short codes. On the other hand, a class of fixed-to-variable length balanced binary line codes was introduced in [12]. This was done by appending a minimum number of bits to each fixed size block of source digits leading to balanced words of variable length. It was shown that this technique is easily implementable and provides high coding efficiency and less redundancy.

A major contribution was made by Imminck and Weber [13] through an efficient encoding of the index prefix for both VL and FL schemes. This scheme is based on distinctly associating each word of a code to a balanced codeword. More details on this method will be provided. Furthermore, the distribution of the prefix length was discussed as well as the complexity of the proposed algorithm.

In this paper, a modification of a scheme described in [13] is proposed to generate efficient and less redundant balanced codes compared to most state-of-the-art techniques. This approach is designed for communication systems that model the data as packets, contrary to cascade-based models. The rest of this paper is structured as follows: a background study is done in Section 2, the system model of the proposed scheme is described in Section 3, and then, in Section 4, the proposed encoding is presented. Sections 5 and 6 provide detailed analysis as well as performance and discussions of the proposed scheme redundancy. Finally, the paper is concluded in Section 7.

2. Background

Let $\mathbf{x} = (x_1x_2\dots x_k)$ be a bipolar sequence of length k and $\mathbf{p} = (p_1p_2\dots p_r)$, the prefix of length r . $\mathbf{c} = (c_1c_2\dots c_n)$ of length $n = k + r$ is the transmitted codeword comprising the encoding of \mathbf{x} denoted as \mathbf{y} and appended to \mathbf{p} , $\mathbf{c} = (\mathbf{p} \cdot \mathbf{y})$. All these words are defined within the alphabet \mathcal{A}^2 where $\mathcal{A}^2 = \{-1, 1\}$. Let $d(\mathbf{x})$ refer to the sum of all digits in \mathbf{x} , also called the disparity of \mathbf{x} . The word \mathbf{x} is said to be balanced if $d(\mathbf{x}) = \sum_{i=1}^k x_i = 0$.

Similarly, the disparity of the first j bits of \mathbf{x} , also called the running digital sum (RDS), is denoted as $d_j(\mathbf{x})$, and $d_j(\mathbf{x}) = \sum_{i=1}^j x_i$, where $1 \leq j \leq k$. For the scope of this paper, the information word length is considered to be even.

2.1. Knuth's balancing scheme

Knuth's celebrated scheme consists of complementing word bits up to certain point. This is equivalent to splitting a word into two segments, where the first one has its bits flipped and the second is unchanged. It was

shown in [7] that this simple and efficient procedure will always generate at least one balanced codeword. If e is the index of the first balancing point then, the disparity of \mathbf{x} is given by:

$$d(\mathbf{x}) = -\sum_{i=1}^e x_i + \sum_{i=e+1}^k x_i. \tag{1}$$

Those summations reflect the two segments that build a balanced codeword. Because $d_{j+1}(\mathbf{x}) = d_j(\mathbf{x}) \pm 2$, it is always achievable to find an index e corresponding to a balancing point such that $d(\mathbf{x}) = 0$. The index e might be unique; by convention, KA only considers the first one while inverting from the least index bits. In a parallel scheme, the index e is encoded as the prefix and prepended to \mathbf{y} . The length of the prefix, r , is given by:

$$r = \lceil \log_2 k \rceil, \text{ for } k \gg 1. \tag{2}$$

The redundancy of a full set of balanced codewords of length k , denoted as $H_0(k)$, equals:

$$H_0(k) = k - \log_2 \binom{k}{k/2}. \tag{3}$$

An approximation of $H_0(k)$ was given in [7] as:

$$H_0(k) \approx \frac{1}{2} \log_2 k + 0.326, \text{ for } k \gg 1. \tag{4}$$

For large k , the Knuth's scheme redundancy is almost twice as large as $H_0(k)$.

2.2. Efficient binary balanced codewords

Let \mathbf{x}^j be the word \mathbf{x} where the first j bits are inverted. If e represents the index of the first balancing point then $\mathbf{y} = \mathbf{x}^e$ is the balanced codeword through Knuth's scheme. There are k different ways of inverting the word \mathbf{x} . In [13], it was established that some words from the set $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^k$ can be associated with the balanced word \mathbf{y} following Knuth's scheme. Let $s(\mathbf{y})$ be the set of all words associated with a balanced codeword, \mathbf{y} , $s(\mathbf{y}) = \{\mathbf{x} : \mathbf{x}^j = \mathbf{y} \text{ with } 1 \leq j \leq k\}$, and $|s(\mathbf{y})|$ its cardinality.

The prefix of the encoded word corresponds to the information word rank within the subset $s(\mathbf{y})$. It was shown in [13] that the size of $s(\mathbf{y})$ is such that $2 \leq |s(\mathbf{y})| \leq \frac{k}{2} + 1$, where $|s(\mathbf{y})| = \max\{d_j(\mathbf{x})\} - \min\{d_j(\mathbf{x})\} + 1$ with $\max\{d_j(\mathbf{x})\}$ and $\min\{d_j(\mathbf{x})\}$ being the maximum and minimum RDS values of \mathbf{x} , respectively. For the FL scheme, the prefix has exactly $\log_2 \left(\frac{k}{2} + 1\right)$ bits, while in the VL scheme, the prefix length varies between 1 and $\log_2 \left(\frac{k}{2} + 1\right)$ bits.

3. System model

Figure 1 shows a model of communication for two different systems. In Figure 1a, the data are received as a set of balanced codewords; in this model, the decoder must keep track of the start and ending of each data block for the purpose of synchronization that relies on prefixes.

In Figure 1b, the packet conception represents a single data block, received one at the time. This concept is used in various communications systems such as Bluetooth/wireless communication, smart grid systems, GSM

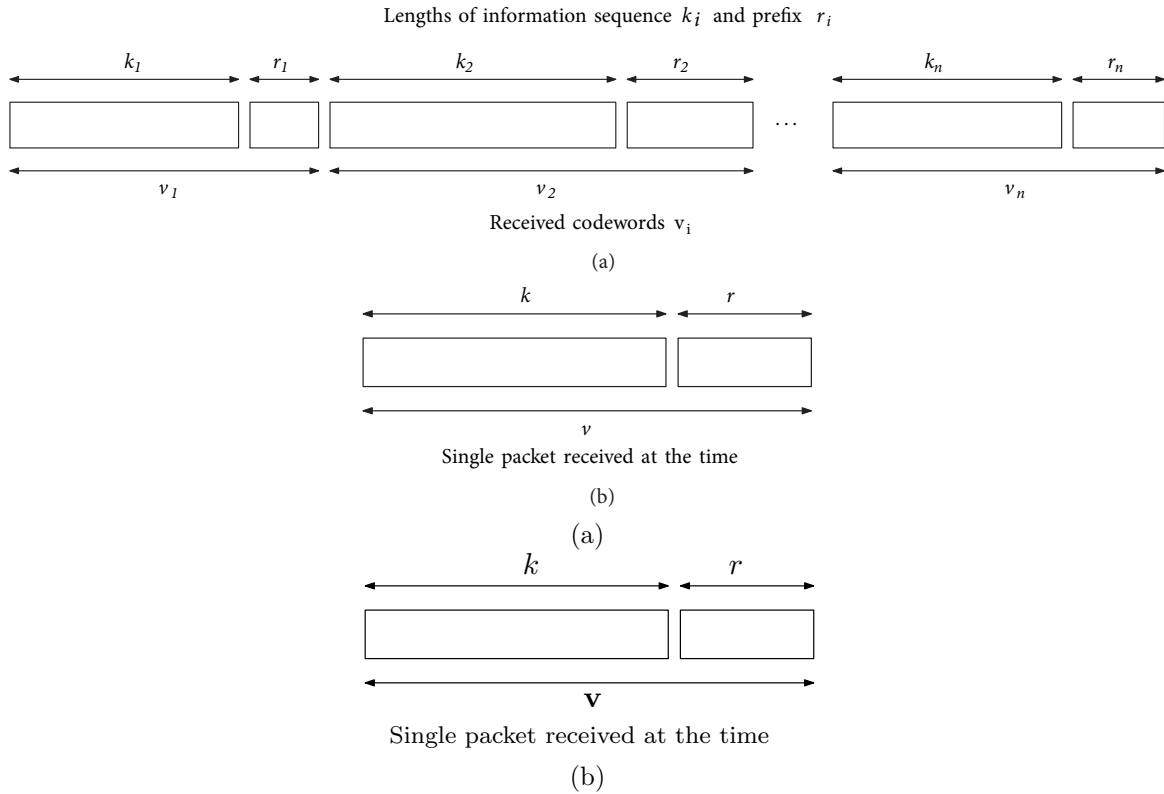


Figure 1. System model: (a) cascade-based vs. (b) packet-based.

networks, power line communication (PLC), visible light communication (VLC), and network communication. The communication is incremental as packets are transmitted one at the time and an “ACK” message is required before a subsequent packet is sent; in this case, the decoder only keeps track at the end of the packet (EOP).

4. Encoding scheme

This consists of associating every information sequence of length k with a balanced codeword using Knuth’s inversion rule as described in [13]. This leads to $\binom{k}{\frac{k}{2}}$ distinct sets. Furthermore, each of these sets is compressed to decrease the redundancy based on Lemma 1.

Let \mathbf{x} be the information sequence to be encoded. If \mathbf{x} is already balanced, a protocol is adopted between the transmitter and the receiver to have a prefix-less codeword; otherwise, \mathbf{x} is balanced following KA, associated with the corresponding balanced codeword \mathbf{y} . Following the same procedure, all information sequences can be associated with balanced codewords, \mathbf{y} , and listed according to the lexicographic order within subsets $s(\mathbf{y})$. The prefix of each \mathbf{x} corresponds to its rank within $s(\mathbf{y})$.

Lemma 1 *Any balanced codeword is always associated with another balanced one.*

Proof This is an observation from the KA structure; any already balanced codeword always results in another balanced one. In the worst-case scenario, the balanced state is always found by inverting all bits of an already balanced codeword. □

Let $|s(\mathbf{y})|$ denote the cardinality of the subset $s(\mathbf{y})$ that comprises all associated sequences with \mathbf{y} . The inclusion of balanced sequences within the set of information sequences as presented in [13] adds an extra rank in the rank position in every set; this is useful for cascade-based models. However, as described in Lemma 1, a balanced sequence is always associated to another one. This important observation leads to the compression of $s(\mathbf{y})$ by removing the already balanced sequence. This is suitable for the packet-based model as described in Figure 1a.

Example 1 Let us consider binary sequences of length $k = 4$.

\mathbf{y}	0011	0101	0110	1001	1010	1100	\mathbf{p}
$s(\mathbf{y})$	①1011	1101	1000	0001	0010	0000	00
	②1111	1001	1110	0111	0110	0100	01
	③ 1100		1010	0101		0011	10

(5)

①1011 → 0011

②1111 → 0111 → 0011

③1100 → 0100 → 0000 → 0010 → 0011

Eq. (5) shows the encoding process described in [13], whereby balanced codewords (marked in bold) are part of subsets $s(\mathbf{y})$. Lines ①, ②, and ③ show how balanced codewords are obtained from Knuth’s balancing scheme. \mathbf{p} represent prefixes.

\mathbf{y}	0011	0101	0110	1001	1010	1100	\mathbf{p}
$s(\mathbf{y})$	①1011	1101	1000	0001	0010	0000	0
	②1111		1110	0111		0100	1

(6)

Eq. (6) presents the proposed encoding process where all subsets do not include balanced sequences.

The cardinality of the subset $s(\mathbf{y})$ can be derived from RDS calculations on the balanced codeword, \mathbf{y} , as presented in Lemma 2.

Lemma 2 $|s(\mathbf{y})| = \max\{d_j(\mathbf{y})\} - \min\{d_j(\mathbf{y})\}$.

Proof It was proved in [13] that $|s(\mathbf{y})| = \max\{d_j(\mathbf{y})\} - \min\{d_k(\mathbf{y})\} + 1$; the balanced codeword was removed from every set. Thus, the new $|s(\mathbf{y})|$ is subtracted by 1, leading to $|s(\mathbf{y})| = \max\{d_j(\mathbf{y})\} - \min\{d_j(\mathbf{y})\}$. □

For any subset $s(\mathbf{y})$, its size is always bounded as per Theorem 1.

Theorem 1 $1 \leq |s(\mathbf{y})| \leq \frac{k}{2}$.

Proof It was established in [13] that $2 \leq |s(\mathbf{y})| \leq \frac{k}{2} + 1$; then, after removing the balanced codeword from every set, it follows that $1 \leq |s(\mathbf{y})| \leq \frac{k}{2}$. □

Therefore, the required fixed prefix length for this scheme is $\log_2 \frac{k}{2}$; this is a slight improvement on Knuth’s scheme that has a redundancy of $\log_2 k$ as well as on the scheme in [13] where it equals $\log_2(\frac{k}{2} + 1)$. In addition, prefixes are obtained from ranking the information sequences associated to a balanced codeword from 0 to $\frac{k}{2} - 1$.

5. Study of sparseness of $|s(\mathbf{y})|$

Let $N(\lambda, k)$ be the number of possible balanced codewords \mathbf{y} of length k such that $|s(\mathbf{y})| = \lambda$.

The following equation holds from Theorem 1:

$$\sum_{\lambda=1}^{k/2} N(\lambda, k) = \binom{k}{\frac{k}{2}}.$$

For the convenience of the reader, details on computing $N(\lambda, k)$ for $1 \leq \lambda \leq \frac{k}{2}$ are derived by following the guidelines in [13].

The derivation of $N(\lambda, k)$ was done using the computation of the number of bipolar sequences whose running sum lies within two finite bounds, $B1$ and $B2$ (with $B2 > B1$) [14].

The interval of running sum values that a sequence may reach, also referred to as the digital sum variation (DSV), is given by $B = B2 - B1 + 1$. Each iteration in the random walk of a sequence defines an entry of a $B \times B$ connection matrix, M_B .

M_B is such that $M_B(i, j) = 1$ if there is a path in the random walk from state s_i to state s_j , and $M_B(i, j) = 0$ if no path can be established. For each iteration, a random walk of the running sum can only move one state up or down. Therefore, $M_B(i + 1, i) = M_B(i, i + 1) = 1$ and $M_B(i, j) = 0$, where $i, j = 1, 2, \dots, B - 1$ as presented in Eq. (7).

$$M_B = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 1 & 0 & 1 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \tag{7}$$

$M_B^k(i, j)$ denotes the (i, j) th entry of the k th power of M_B .

Theorem 2 *The number of balanced codewords \mathbf{y} of length k and $|s(\mathbf{y})| = \lambda$, $N(\lambda, k)$ for $1 \leq \lambda \leq \frac{k}{2}$, is such that*

$$N(\lambda, k) = \sum_{i=1}^{\lambda+1} M_{\lambda+1}^k(i, i) - 2 \sum_{i=1}^{\lambda} M_{\lambda}^k(i, i) + \sum_{i=1}^{\lambda-1} M_{\lambda-1}^k(i, i).$$

Proof The number of balanced codewords such that $|s(\mathbf{y})| = \lambda'$ for $2 \leq \lambda' \leq \frac{k}{2} + 1$ in [13] was

$$N(\lambda', k) = \sum_{i=1}^{\lambda'} M_{\lambda'}^k(i, i) - 2 \sum_{i=1}^{\lambda'-1} M_{\lambda'-1}^k(i, i) + \sum_{i=1}^{\lambda'-2} M_{\lambda'-2}^k(i, i).$$

Therefore, for $1 \leq \lambda \leq \frac{k}{2}$, the set of all random walks between bounds $B2$ and $B1$ is shifted one unit down. This leads to $N(\lambda, k)$ balanced codewords where $\lambda = \lambda' - 1$.

This leads to the following:

$$N(\lambda, k) = \sum_{i=1}^{\lambda+1} M_{\lambda+1}^k(i, i) - 2 \sum_{i=1}^{\lambda} M_{\lambda}^k(i, i) + \sum_{i=1}^{\lambda-1} M_{\lambda-1}^k(i, i).$$

□

A simplified expression of M_B was provided in [13] based on a formula to compute powers of M_B derived by Salkuyeh [15] as follows:

$$\sum_{i=1}^B M_B^k(i, i) = 2^k \sum_{i=1}^B \cos^k \frac{\pi i}{B+1}. \tag{8}$$

This makes the computation of $N(\lambda, k)$ much simpler, as follows:

$$N(\lambda, k) = 2^k \left(\sum_{i=1}^{\lambda+1} \cos^k \frac{\pi i}{\lambda+2} - 2 \sum_{i=1}^{\lambda} \cos^k \frac{\pi i}{\lambda+1} + \sum_{i=1}^{\lambda-1} \cos^k \frac{\pi i}{\lambda} \right). \tag{9}$$

The computation of $N(\lambda, k)$ as presented in Eq. (9) becomes obvious for special values of λ as shown in Eq. (10). The enumeration of sequences corresponding to these values of λ as well as the pseudocode for computing $|s(\mathbf{y})|$, for generating the ordered set of information sequences and determining the prefix index, were provided in [13].

λ	$N(\lambda, k)$	(10)
1	2	
2	$2(2^{\frac{k}{2}-1})$	
$\frac{k}{2} - 1$	$k(k-4), k > 4$	
$\frac{k}{2}$	k	

6. Analysis and discussions

In this section, the average number of bits denoted as $H(k)$ required to encode the prefix index of a sequence of length k is computed. The number of all information sequences associated with balanced codewords is $2^k - \binom{k}{\frac{k}{2}}$.

$$\sum_{\lambda=1}^{k/2} \lambda N(\lambda, k) = 2^k - \binom{k}{\frac{k}{2}}. \tag{11}$$

It follows that

$$H(k) = \frac{\sum_{\lambda=1}^{k/2} \lambda N(\lambda, k) \log_2 \lambda}{2^k - \binom{k}{\frac{k}{2}}}. \tag{12}$$

The average number of bits for the construction in [13] is as follows:

$$H_1(k) = 2^{-k} \sum_{\lambda=2}^{\frac{k}{2}+1} \lambda N(\lambda, k) \log_2 \lambda. \tag{13}$$

The average number of bits for the method in [9] is given by

$$H_2(k) = \sum_{c=1}^{\frac{k}{2}} P(c) AV(c), \tag{14}$$

where

$$P(c) = 2^{c+1-k} \binom{k-1-c}{\frac{k}{2}-c}, 1 \leq c \leq \frac{k}{2}, d = c - 2^{\lfloor \log_2 c \rfloor},$$

and

$$AV(c) = (c - 2d) \cdot \lceil \log_2 c \rceil \cdot \frac{1}{2^{\lfloor \log_2 c \rfloor}} + 2d \cdot \frac{1}{2^{\lceil \log_2 c \rceil}} \cdot \lceil \log_2 c \rceil.$$

Table 1 presents the comparison of the average number of bits necessary to encode the prefix from various schemes. Let d_{H_a, H_b} be the difference between the average prefix length H_a and H_b ; we observed that $d_{H, H_0} \leq 0.61$, $d_{H, H_1} \leq 0.64$, and $d_{H_2, H} \leq 1.23$.

Table 1. Comparison of the prefix’s average number of bits

k	H_0	H	H_1	H_2
4	1.4150	0.8000	1.4387	0.5000
8	1.8707	1.4632	1.8985	0.9375
16	2.3483	2.0806	2.3790	1.3706
32	2.8370	2.6629	2.8691	1.8082
64	3.3314	3.2207	3.3641	2.2516
128	3.8286	3.7615	3.8616	2.7039
256	4.3272	4.2902	4.3603	3.1647
512	4.8265	4.8104	4.8597	3.6330
1024	5.3261	5.3246	5.3594	4.1082

Figure 2 shows the comparison between the average redundancy for balanced prefixes for $H(k)$ and $H_1(k)$, denoted as $H'(k)$ and $H'_1(k)$ respectively, as well as $\log_2(k)$ and $\lceil \log_2(k) \rceil$. $H'(k)$ is obtained from a simple modification of $H(k)$ provided in Eq. (12) as follows:

$$H'(k) = \frac{\sum_{\lambda=1}^{k/2} \lambda N(\lambda, k) \Delta(\lambda)}{2^k - \binom{k}{\frac{k}{2}}}. \tag{15}$$

Similarly, $H'_1(k)$ is derived from $H_1(k)$ given in Eq. (13) as follows:

$$H'_1(k) = 2^{-k} \sum_{\lambda=2}^{\frac{k}{2}+1} \lambda N(\lambda, k) \Delta(\lambda), \tag{16}$$

where $\Delta(\lambda)$ corresponds to the smallest value of length k such that $\binom{k}{\frac{k}{2}} \geq \lambda$.

The graphs of $\log_2(k)$ and $\lceil \log_2(k) \rceil$ represents the minimum redundancy and that of integer valued redundancy of the traditional Knuth’s construction. We observe that it is only from $k > 64$ that the average redundancy of the scheme presented in [13] is less than that of Knuth’s scheme, whereas for the proposed construction, the average redundancy becomes advantageous as soon as $k > 16$. Furthermore, the proposed scheme outperforms [13], at least for $k < 1024$.

According to Theorem 1, the two coding schemes are applicable for the proposed scheme. For the FL prefix construction, the encoding of the prefix requires exactly $\log_2(\frac{k}{2})$ bits representing the balanced index e ranging from 0 to $\frac{k}{2} - 1$, whereas for the VL scheme, the prefix length varies between 0 and $\log_2(\frac{k}{2})$ depending on the nature of the information to be encoded. A zero prefix is used when the information sequence is already balanced. However, the VL scheme is more efficient than the FL one on the average basis.

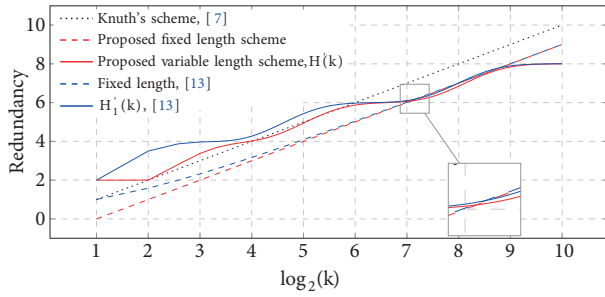


Figure 2. Redundancy comparison for various schemes.

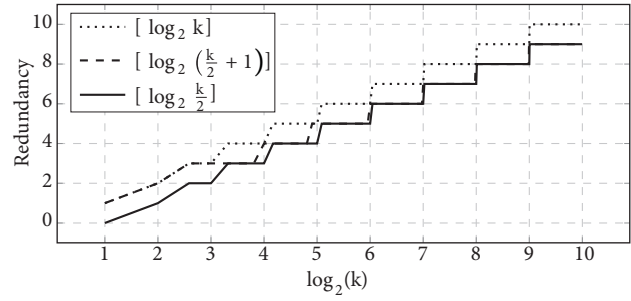


Figure 3. Rounded up FL prefix schemes.

For practical systems purposes, a redundancy length r can only be a positive integer value. For [13], $r = \lceil \log_2(\frac{k}{2} + 1) \rceil$; in [7], $r = \lceil \log_2(k) \rceil$; and for the proposed construction, $r = \lceil \log_2(k) \rceil$. Figure 3 presents the rounded up FL prefix schemes. This shows that the proposed FL prefix scheme is more efficient than that of [13] for at least $k < 512$. This improvement on short length is a great advantage for most communication systems as they make use of short packet sizes to convey information through various channels to avoid latency.

7. Overall balancing

With the objective of deviating from the traditional lookup tables, for which memory is consuming in coding, we propose a 4B6B coding based on KA. Table 2 presents our proposed 4B6B coding, which does not make use of lookup tables; red digits represent the inverted portion and bold digits are the positions of the balancing point index. We can notice that the table is divided into two parts: the first one consists of all inputs starting with a ‘0’ having corresponding balanced codewords starting with a ‘1’ and the second is all inputs starting with a ‘1’ and having their match balanced codewords starting with a ‘0’.

Table 2. Proposed RLL 4B6B based on KA.

Input	Balanced	Input	Balanced
0000	1100 10	0100	1100 01
0001	1001 01	0101	1001 10
0010	1010 01	0110	1010 10
0011	1101 00	0111	1000 11

Input	Balanced	Input	Balanced
1000	0111 00	1100	0010 11
1001	0101 10	1101	0101 01
1010	0110 10	1110	0110 01
1011	0011 01	1111	0011 10

The encoder prefix consists of encoding every 4 digits into 6; a prefix with length different from multiples of 4 should be filled up with ‘0’s, which leads to a FL scheme. The input word is inverted up to a certain index, which is appended at the end of the word according to the following rules, which are embedded in the decoder: if the input starts with a ‘0’, the index positions, e with $1 \leq e \leq 4$, are as follows: $1 \rightarrow 01$, $2 \rightarrow 10$, $3 \rightarrow 00$, and $4 \rightarrow 11$. For an input starting with a ‘1’, the index positions are $1 \rightarrow 01$, $2 \rightarrow 01$, $3 \rightarrow 11$, and $4 \rightarrow 00$.

Therefore, an overall balancing can be achieved by encoding the prefix through the proposed 4B6B coding from Table 2.

8. Conclusion

A modification of the construction given in [13] was proposed for packet transmission systems. The proposed scheme requires exactly $\log_2(\frac{k}{2})$ bits for the FL prefix and a prefix length between 0 and $\log_2(\frac{k}{2})$ bits for the VL scheme. The sparseness of $|s(\mathbf{y})|$ was studied and the average efficiency of this scheme was discussed and compared to existing ones. The proposed construction is more efficient and less redundant than various schemes; it does not make use of lookup tables or enumerative coding. Future works include a further compression of the prefix length for overall balancing through advanced efficient constructions such as in [16–18]. On the other hand, we can extend the proposed algorithm to investigate the efficient balancing of q -ary sequences as higher alphabets, especially powers of twos, as they present various advantages in communication systems in terms of reducing latency, improving communication speed, and increasing robustness and reliability [19–22].

Acknowledgments

The authors would like to acknowledge Jos Weber for proofreading this article and for constructive discussions. This work was supported partially by the Global Excellence Stature program.

References

- [1] Immink K. Coding methods for high-density optical recording, Philips Journal of Research 1986; 41 (4): 410-430.
- [2] Leiss E. Data integrity in digital optical disks. IEEE Transactions on Computers 1984; 33 (9): 818-827. doi: 10.1109/TC.1984.1676498
- [3] Al-Bassam S, Bose B. Design of efficient error-correcting balanced codes. IEEE Transactions on Computers 1993; 42 (10): 1261-1266. doi: 10.1109/12.257712
- [4] Weber J, Immink K, Ferreira H. Error-correcting balanced Knuth codes. IEEE Transactions on Information Theory 2012; 58 (1): 82-89. doi: 10.1109/TIT.2011.2167954
- [5] Cattermole K. Principles of digital line coding. International Journal of Electronics 1983; 55 (1): 3-33. doi: 10.1080/00207218308961573
- [6] Tabor J. Noise Reduction Using Low Weight and Constant Weight Coding Techniques. Cambridge, MA, USA: MIT Computer Science and Artificial Intelligence Lab, 1990.
- [7] Knuth D. Efficient balanced codes. IEEE Transactions on Information Theory 1986; 32 (1): 51-53. doi: 10.1109/TIT.1986.1057136
- [8] Weber J, Immink K. Knuth's balanced codes revisited. IEEE Transactions on Information Theory 2010; 56 (4): 1673-1679. doi: 10.1109/TIT.2010.2040868
- [9] Al-Rababa'a A, Dubé D, Chouinard J. Using bit recycling to reduce Knuth's balanced codes redundancy. In: 2013 13th Canadian Workshop on Information Theory; Toronto, Canada; 2013. pp. 6-11.
- [10] Dubé D, Mechqrane M. Almost minimum-redundancy construction of balanced codes using limited-precision integers. In: 2017 15th Canadian Workshop on Information Theory; Quebec City, Canada; 2017. pp. 1-5.
- [11] Swart T, Weber J. Binary variable-to-fixed length balancing scheme with simple encoding/decoding. IEEE Communications Letters 2018; 22 (10): 1992-1995. doi: 10.1109/LCOMM.2018.2865350
- [12] Rocha V, Lemos-Neto J, Pacheco A. Class of easily implementable fixed-length to variable-length balanced binary line codes. Electronics Letters 2019; 55 (5): 266-268. doi: 10.1049/el.2018.7032

- [13] Immink K, Weber J. Very efficient balanced codes. *Journal on Selected Areas in Communications* 2010; 28 (2): 188-192. doi: 10.1109/JSAC.2010.100207
- [14] Chien T. Upper bound on the efficiency of DC-constrained codes. *Bell System Technical Journal* 1970; 49 (9): 2267-2287. doi: 10.1002/j.1538-7305.1970.tb02525.x
- [15] Salkuyeh D. Positive integer powers of the tri-diagonal Toeplitz matrices. *International Mathematical Forum* 2006; 1 (22): 1061-1065.
- [16] Mambou EN, Swart T. Encoding and decoding of balanced q -ary sequences using a Gray code prefix. In: 2016 IEEE International Symposium on Information Theory; Barcelona, Spain; 2016. pp. 380-384.
- [17] Mambou EN, Swart T. A construction for balancing non-binary sequences based on Gray code prefixes. *IEEE Transactions on Information Theory* 2017; 64 (8): 5961-5969. doi: 10.1109/TIT.2017.2766668
- [18] Mambou EN, Swart T. Construction of q -ary constant weight sequences using a Knuth-like approach. In: 2017 IEEE International Symposium of Information Theory; Aachen, Germany; 2017. pp. 2028-2032.
- [19] Ulrich W. Non-binary error correction codes. *Bell System Technical Journal* 1957; 36 (6): 1341-1388. doi: 10.1002/j.1538-7305.1957.tb01514.x
- [20] Berrou C, Jezequel M, Douillard C, Kerouedan S. The advantages of non-binary turbo codes. In: *Proceedings 2001 IEEE Information Theory Workshop*; Cairns, Australia; 2001. pp. 61-63.
- [21] Pfleschinger S, Declercq D. Getting closer to MIMO capacity with non-binary codes and spatial multiplexing. In: 2010 IEEE Global Telecommunication Conference; Miami, FL, USA; 2010. pp. 1-5.
- [22] Karzand M, Telatar E. Polar codes for q -ary source coding. In: 2010 IEEE International Symposium on Information Theory; Austin, TX, USA; 2010. pp. 909-912.