Extension of Knuth's Balancing Algorithm with Error Correction*

Jos H. Weber	Kees A. Schouhamer Immink	Hendrik C. Ferreira
Delft Univ. of Techn.	Turing Machines BV	Univ. of Johannesburg
Mekelweg 4	Willemskade 15b-d	Dept. E&E Eng. Sc.
2628 CD Delft	3016 DK Rotterdam	Auckland Park 2006
The Netherlands	The Netherlands	South Africa
j.h.weber@tudelft.nl	immink@turing-machines.com	hcferreira@uj.ac.za

Abstract

Knuth's celebrated balancing method consists of inverting the first z bits in a binary information sequence, such that the resulting sequence has as many ones as zeroes, and communicating the index z to the receiver through a short balanced prefix. In the proposed method, Knuth's scheme is extended with error-correcting capabilities, where it is allowed to give unequal protection levels to the prefix and the payload. An analysis with respect to the redundancy of the proposed method is performed, showing good results while maintaining the simplicity features of the original scheme.

1 Introduction

Sets of binary sequences that have a fixed length n and a fixed weight w (number of ones) are usually called *constant-weight codes*. An important sub-class is formed by the so-called *balanced codes*, for which n is even and $w = n/2$, i.e., all codewords have as many zeroes as ones. Such codes have found application in various transmission and (optical/magnetic) recording systems. A survey on balanced codes can be found in [6].

A simple method for generating balanced codewords, which is capable of encoding and decoding (very) large blocks, was proposed by Knuth [4] in 1986. In his method, an m-bit binary data word, m even, is forwarded to the encoder. The encoder inverts the first z bits of the data word, where z is chosen in such a way that the modified word has equal numbers of zeroes and ones. Knuth showed that such an index z can always be found. The index z is represented by a balanced word of length p. The p-bit prefix word followed by the modified m-bit data word are both transmitted, so that the rate of the code is $m/(m+p)$. The receiver can easily undo the inversion of the first z bits received once z is computed from the prefix. Both encoder and decoder do not require large look-up tables, and Knuth's algorithm is therefore very attractive for constructing long balanced codewords. The redundancy of Knuth's method is roughly twice the redundancy of a code which uses the full set of balanced words. Since the latter has a prohibitively high complexity in case of large lengths, the factor of two can be considered as a price to be paid for simplicity. In [7] and [9], modifications to Knuth's method are presented closing this gap while maintaining sufficient simplicity.

Knuth's method does not provide protection against errors which may occur during transmission or storage. Actually, errors in the prefix may lead to catastrophic error propagation in the data word. Here, we propose and analyze a method to extend Knuth's original scheme with error correcting capabilities. Previous constructions for error-correcting balanced codes were given in [8], [2] and [5]. In [8], van Tilborg and

*This project was supported by grant *Theory and Practice of Coding and Cryptography*, Award Number: NRF-CRP2-2007-03. Kees A. Schouhamer Immink is also with the Nanyang Technological University, Singapore.

Blaum introduced the idea to consider short balanced blocks as symbols of an alphabet and to construct error-correcting codes over that alphabet. Only moderate rates can be achieved by this method, but it has the advantage of limiting the digital sum variation and the runlengths. In [2], Al-Bassam and Bose constructed balanced codes correcting a single error, which can be extended to codes correcting up to two, three, or four errors by concatenation techniques. In [5], Mazumdar, Roth, and Vontobel considered linear balancing sets and applied such sets to obtain error-correcting coding schemes in which the codewords are balanced. In the method proposed in the current paper, we stay very close to the original Knuth algorithm. Hence, we only operate in the binary field and inherit the low-complexity features of Knuth's method. In our method, the error-correcting capability can be any number. The focus will be on long codes, for which table look-up methods are unfeasible. An additional feature is the possibility to assign different error protection levels to the prefix and the payload, which could be useful when designing the scheme to achieve a certain required error performance while optimizing the rate.

The rest of this paper is organized as follows. In Section 2, the proposed method for providing balancing and error-correcting capabilities is presented. In Section 3, the redundancy of the new scheme is considered. Finally, the results of this paper are discussed in Section 4.

2 Construction Method

The proposed construction method is based on a combination of conventional error correction techniques and Knuth's method for obtaining balanced words. The encoding procedure consists of fours steps which are described below and illustrated in Figure 1. The input to the encoder is a binary data block $\mathbf{u}$ of length k. Let b^i denote a run of i bits b, e.g., $1^3 0^5 = 11100000$.

1. Encode $\mathbf{u}$ using a binary linear (m, k, d_1) block code $\mathcal{C}_1$ of dimension k, Hamming distance d_1, and even length m. The encoding function is denoted by ϕ.

2. Find a balancing index z for the obtained codeword $\phi(\mathbf{u})$, with $1 \leq z \leq m$.

3. Invert the first z bits of $\phi(\mathbf{u})$, resulting in the balanced word $\mathbf{c} = \phi(\mathbf{u}) + 1^z 0^{m-z}$.

4. Encode the number z into a unique codeword $\mathbf{s}$ from a binary code $\mathcal{C}_2$ of even length p, constant weight $p/2$, and Hamming distance d_2. The encoding function is denoted by ψ.

The output of the encoder is the concatenation of the balanced word $\mathbf{s} = \psi(z)$, called the *prefix*, and the balanced word $\mathbf{c} = \phi(\mathbf{u}) + 1^z 0^{m-z}$, called the *bulk* or *payload*. It is obvious that the resulting code $\mathcal{C}$ is balanced and has length $n = m + p$ and redundancy $r = m + p - k$, and thus code rate $R = k/(m + p)$ and normalized redundancy $\rho = 1 - R = 1 - k/(m+p)$. Its Hamming distance d satisfies the following lower bound.

Theorem 1 *The Hamming distance d of code $\mathcal{C}$ is at least*

$$\min\{2\lceil d_1/2 \rceil, d_2\}.$$

Proof: Let $(\mathbf{s}, \mathbf{c})$ and $(\mathbf{s}', \mathbf{c}')$ denote two different codewords of $\mathcal{C}$ and let $z = \psi^{-1}(\mathbf{s})$. If $\mathbf{s} \neq \mathbf{s}'$, then the Hamming distance between the codewords is at least d_2, since $\mathbf{s}$ and $\mathbf{s}'$ are both in $\mathcal{C}_2$. If $\mathbf{s} = \mathbf{s}'$, then the Hamming distance between the codewords is at least $2\lceil d_1/2 \rceil$, which follows from the fact that $\mathbf{c} + 1^z 0^{m-z}$ and $\mathbf{c}' + 1^z 0^{m-z}$ are two different codewords from $\mathcal{C}_1$, implying that $d_{\mathrm{H}}(\mathbf{c}, \mathbf{c}') \geq d_1$, and the fact that $\mathbf{c}$ and $\mathbf{c}'$ are both balanced, implying that $d_{\mathrm{H}}(\mathbf{c}, \mathbf{c}')$ is even. ■

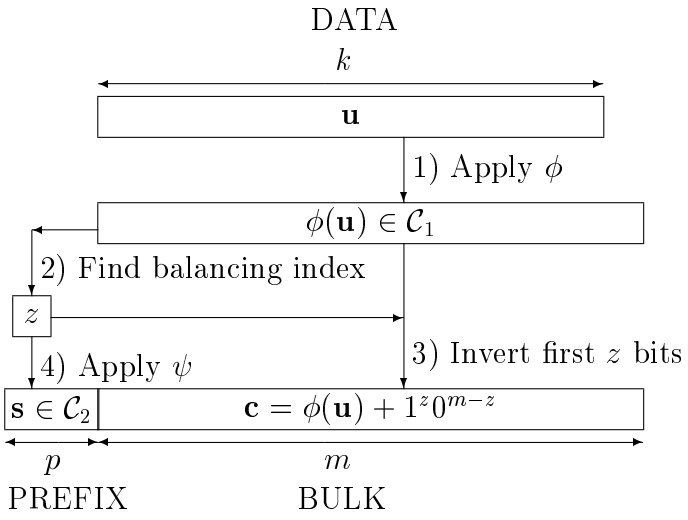

Figure 1: Encoding procedure.

Corollary 1 *In order to make $\mathcal{C}$ capable of correcting up to t errors, it suffices to choose constituent codes $\mathcal{C}_1$ and $\mathcal{C}_2$ with distances $d_1 = 2t + 1$ and $d_2 = 2t + 2$, respectively.*

Example 1 *Let the information block length be chosen as $k = 750$. We consider $(750 + 10t_1, 750, 2t_1 + 1)$ codes $\mathcal{C}_1$, with $t_1 = 0, 1, \ldots, 4$, obtained by shortening $(1023, 1023 - 10t_1, 2t_1 + 1)$ BCH codes. For $\mathcal{C}_2$ we consider the shortest known balanced codes with cardinality at least $750 + 10t_1$ and Hamming distance $2t_2 + 2$, with $t_2 = 0, 1, \ldots, 4$. Such balanced codes are tabulated on [3], from which we collected the cardinalities of some short codes in Table 1.*

An overview of the parameters of codes obtained by choosing $t_1 = t_2 = t$ is provided in Table 2. If $t = 0$, then it is found from Table 1 that a prefix of length 12 is required to represent the 750 possible balancing positions without error correction capabilities, as in the original Knuth case, leading to a code rate of $750/(750 + 12) = 0.9843$, i.e., a normalized redundancy of $1 - 0.9843 = 0.0157$. If $t = 1$, then it is found from Table 1 that a prefix of length 16 is required to represent the 760 possible balancing positions in the BCH codeword with a single error correction capability, leading to a higher normalized redundancy of $1 - 750/(760 + 16) = 0.0335$. Further increasing the value of t leads to higher distances at the expense of higher redundancies, as can be checked from the table.

The proposed scheme also offers the option to provide unequal error protection to the bulk and the prefix. An overview of the parameters of codes obtained by fixing $t_1 = 3$ and varying t_2 is provided in Table 3. Choosing $t_2 = 0$, i.e., providing no error correction capability to the prefix, gives a Hamming distance $d = 2$ and a normalized redundancy $1 - 750/(780 + 12) = 0.0530$. Note that increasing t_2 up to the value of 3 increases both the Hamming distance (since $d = \min\{8, 2t_2 + 2\} = 2t_2 + 2$) and the redundancy. However, also note that further increasing t_2 from 3 to 4 (or beyond) increases the redundancy, without the reward of an improved distance d (since it is stuck at $d = 8$ due to the fact that $t_1 = 3$). ■

Table 1: Cardinalities of the largest known balanced codes with length $p \leq 28$ and Hamming distance $d_2 \leq 10$ [3].

p	$d_2 = 2$	$d_2 = 4$	$d_2 = 6$	$d_2 = 8$	$d_2 = 10$
2	2				
4	6	2			
6	20	4	2		
8	70	14	2	2	
10	252	36	6	2	2
12	924	132	22	4	2
14	3432	325	42	8	2
16	12870	1170	120	30	4
18	48620	3540	320	48	10
20	184756	13452	944	176	38
22	705432	40624	2636	672	46
24	2704156	151484	5616	2576	123
26	10400600	431724	16117	3588	210
28	40116600	1535756	53021	6218	790

Table 2: Code parameters in the setting of Example 1 for the case $t_1 = t_2 = t$ with $0 \leq t \leq 4$.

	$\mathcal{C}_1$			$\mathcal{C}_2$		$\mathcal{C}$		
t	m	k	d_1	p	d_2	n	ρ	d
0	750	750	1	12	2	762	0.0157	2
1	760	750	3	16	4	776	0.0335	4
2	770	750	5	20	6	790	0.0506	6
3	780	750	7	24	8	804	0.0672	8
4	790	750	9	28	10	818	0.0831	10

Table 3: Code parameters in the setting of Example 1 for the case $t_1 = 3$ and $0 \leq t_2 \leq 4$.

		$\mathcal{C}_1$			$\mathcal{C}_2$		$\mathcal{C}$		
t_1	t_2	m	k	d_1	p	d_2	n	ρ	d
3	0	780	750	7	12	2	792	0.0530	2
3	1	780	750	7	16	4	796	0.0578	4
3	2	780	750	7	20	6	800	0.0625	6
3	3	780	750	7	24	8	804	0.0672	8
3	4	780	750	7	28	10	808	0.0718	8

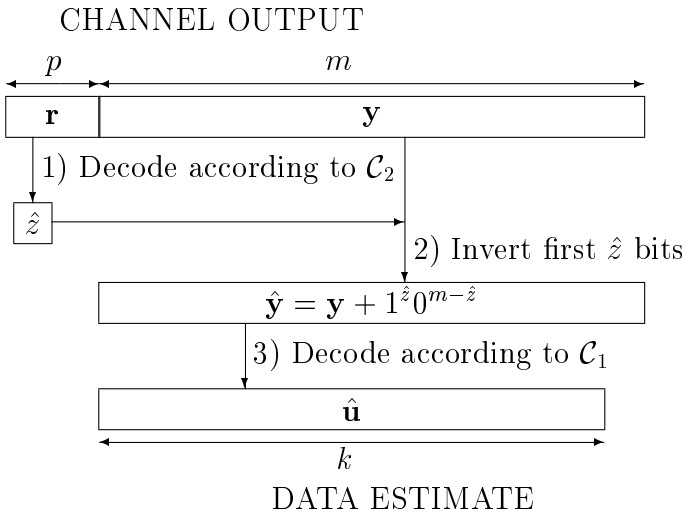

Figure 2: Decoding procedure.

Upon receipt of a sequence $(\mathbf{r}, \mathbf{y})$, where $\mathbf{r}$ and $\mathbf{y}$ have lengths p and m, respectively, a simple decoding procedure, illustrated in Figure 2, consists of the following steps.

1. Look for a codeword $\mathbf{q}$ in $\mathcal{C}_2$ which is closest to $\mathbf{r}$, and set $\hat{z} = \psi^{-1}(\mathbf{q})$.
2. Invert the first $\hat{z}$ bits in $\mathbf{y}$, i.e., set $\hat{\mathbf{y}} = \mathbf{y} + 1^{\hat{z}}0^{m-\hat{z}}$.
3. Decode $\hat{\mathbf{y}}$ according to a decoding algorithm for code $\mathcal{C}_1$, leading to an estimated codeword $\hat{\mathbf{c}}$ and thus to an estimated information block $\hat{\mathbf{u}} = \phi^{-1}(\hat{\mathbf{c}})$.

The following results are immediate.

Theorem 2 *The proposed decoding procedure for code $\mathcal{C}$ corrects any error pattern with at most $d_2/2-1$ errors in the first p bits and at most $\lceil d_1/2 \rceil - 1$ errors in the last m bits.*

Corollary 2 *The proposed decoding procedure for code $\mathcal{C}$ corrects up to the number of errors*

$$\min\{\lceil d_1/2 \rceil - 1, d_2/2 - 1\}$$

guaranteed by the Hamming distance result from Theorem 1.

Corollary 3 *The proposed decoding procedure for code $\mathcal{C}$ corrects up to t errors if $d_1 = 2t + 1$ and $d_2 = 2t + 2$.*

In the next section, we study the redundancy of the proposed method, which we compare with the minimum redundancy required to achieve balancing in combination with error correction capabilities.

3 Redundancy

Let $A(n, d, w)$ denote the maximum cardinality of a code of length n, constant weight w, and even Hamming distance d. Hence, for any balanced code of even length n and Hamming distance d, the redundancy is at least

$$r_{\min} = n - \log_2 A(n, d, n/2). \tag{1}$$

Since

$$A(n, 2, n/2) = \binom{n}{n/2}, \tag{2}$$

the minimum redundancy for a balanced code without error correction capabilities [4] is

$$\begin{aligned} r_0 &= n - \log_2 \binom{n}{n/2} && (3)\\ &\approx \frac{1}{2}\log_2 n + \frac{1}{2}\log_2(\pi/2) && (4)\\ &\approx \frac{1}{2}\log_2 n + 0.326, && (5) \end{aligned}$$

where the first approximation is due to the well-known Stirling formula

$$n! \approx \sqrt{2\pi n} n^n e^{-n}. \tag{6}$$

No general expression for $A(n, d, w)$ is known, but bounds are available in literature. From Theorem 12 in [1], we have the upper bound

$$A(n, d, n/2) \leq \frac{\binom{n}{n/2-t}}{\binom{n/2}{t}} = \frac{\binom{n}{n/2}}{\binom{n/2+t}{t}}, \tag{7}$$

where $t = d/2 - 1$. Note that for $d = 2$, i.e., $t = 0$, this gives the same expression as (2), and thus the bound is tight in this case. The upper bound (7) can be used to lower bound the minimum redundancy in case $t \geq 1$, i.e.,

$$\begin{aligned} r_{\min} &\geq n - \log_2 \binom{n}{n/2} + \log_2 \binom{n/2+t}{t} && (8)\\ &\approx \left(t + \frac{1}{2}\right)\log_2\left(\frac{n}{t}\right) + t(\log_2 e - 1) - 1, && (9) \end{aligned}$$

where the inequality follows from (1) and (7) and the approximation is due to Stirling's formula. The lower bound from (8) will be denoted by $r^*_{\min}$. We see that it consists of the sum of a contribution r_0 from the balance property and a contribution $\log_2 \binom{n/2+t}{t}$ from the capability of correcting up to t errors.

Next, we will investigate the difference between the lower bound $r^*_{\min}$ on the redundancy, of which it is unknown whether it is achievable in general, and the redundancy r of the proposed construction method. We know from [4] that the redundancy of the Knuth scheme, without error correction, falls a factor of two short of the minimum achievable redundancy. Obviously, this is a price to be paid for simplicity. For the example introduced in the previous section, we get the following results when error correction is involved.

Table 4: Redundancy comparison in the setting of Example 2.

t	n	$\rho = 1 - 750/n$	$\rho^*_{\min} = r^*_{\min}/n$	$\rho/\rho^*_{\min}$
0	762	0.0157	0.0067	2.34
1	776	0.0335	0.0177	1.89
2	790	0.0506	0.0271	1.87
3	804	0.0672	0.0355	1.89
4	818	0.0831	0.0432	1.92

Example 2 *Consider again the setting from Example 1 and choose* $t_1 = t_2 = t$*, with* $t = 0, 1, \ldots, 4$*. Hence,* $\mathcal{C}_1$ *has length* $m = 750 + 10t$ *and Hamming distance* $2t + 1$*, while* $\mathcal{C}_2$ *has length* $p = 12 + 4t$ *and Hamming distance* $2t + 2$*. Thus, the code* $\mathcal{C}$ *has length* $n = m + p = 762 + 14t$*, redundancy* $r = 12 + 14t$*, normalized redundancy*

$$\rho = \frac{12 + 14t}{762 + 14t},$$

and Hamming distance $d = 2t + 2$*.*

In Table 4, we compare the normalized redundancy ρ *to* $\rho^*_{\min}$*. Note that* $\rho/\rho^*_{\min}$ *is, as for Knuth's original method, close to 2, in fact a little bit less in case we have error-correcting capabilities. The factor* $\rho/\rho_{\min}$*, the price to be paid for simplicity, may even be smaller since* $\rho^*_{\min}$ *is only a lower bound on* $\rho_{\min}$ *of which it is unknown whether it is achievable in case* $t \geq 1$*.* ■

In general, the redundancy of the proposed method is equal to the sum of the redundancy of code $\mathcal{C}_1$ and the length of the prefix. For neither of these terms a general expression is available. The former depends on the choice of $\mathcal{C}_1$. For example, for BCH codes it is roughly $t \log_2 m$. The latter, i.e., the length of the prefix, can be decomposed into two parts: a contribution of length $\log_2 m$ identifying the balancing index and a contribution of length roughly $(t + 1/2) \log_2 \log_2 m$ (based on the presented bounds on constant weight codes) providing the error correction and balancing properties to the prefix. Hence, the total redundancy of the proposed method can be approximated as

$$(t + 1) \log_2 m + (t + 1/2) \log_2 \log_2 m.$$

In comparison, it follows from (9) that the minimum redundancy for any balanced code with large length n and small error correction t capability is approximately

$$(t + 1/2) \log_2 n.$$

Although the results presented in this analysis are based on bounds and approximations rather than exact results, it seems safe to conclude that in general, as in the example, the redundancy of the presented method is within a factor of two of the optimum. The redundancy of the codes presented in [2] is (slightly) lower, but the method presented here is simpler and more general (since the constructions from [2] are for $t \leq 4$ only). The constructions from [8] are of a completely different nature, with much higher redundancies but balancing being established on a very small block scale.

4 Discussion

We have extended Knuth's balancing scheme with error-correcting capabilities. The approach is very general in the sense that any block code can be used to protect the payload, while the prefix of length p is protected by a constant-weight code where the weight is $p/2$. As for the original Knuth algorithm, the scheme's simplicity comes at the price of a somewhat higher redundancy than the most efficient but prohibitively complex code. In [10], it is demonstrated that, in order to meet a certain target block or bit error probability in an efficient way, the distances of the constituent codes may preferably be unequal. Hence, from the performance perspective, the overall Hamming distance is of minor importance.

In conclusion, the proposed scheme is an attractive simple alternative to achieve (long) balanced sequences with error correction properties.

References

[1] E. Agrell, A. Vardy, and K. Zeger, "Upper Bounds for Constant-Weight Codes", *IEEE Trans. Inform. Theory*, vol. 46, no. 7, pp. 2373-2395, November 2000.

[2] S. Al-Bassam and B. Bose, "Design of Efficient Error-Correcting Balanced Codes", *IEEE Trans. Computers*, vol. 42, no. 10, pp. 1261-1266, October 1993.

[3] A.E. Brouwer, "Bounds for Binary Constant Weight Codes", `http://www.win.tue.nl/~aeb/codes/Andw.html`.

[4] D.E. Knuth, "Efficient Balanced Codes", *IEEE Trans. Inform. Theory,* vol. IT-32, no. 1, pp. 51-53, Jan. 1986.

[5] A. Mazumdar, R.M. Roth, and P.O. Vontobel, "On Linear Balancing Sets", IEEE International Symposium on Information Theory, Seoul, South Korea, pp. 2699-2703, June-July 2009.

[6] K.A. Schouhamer Immink, *Codes for Mass Data Storage Systems,* Second Edition, Shannon Foundation Publishers, Eindhoven, The Netherlands, 2004.

[7] K.A. Schouhamer Immink and J.H. Weber, "Very Efficient Balanced Codes", *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 2, pp. 188-192, February 2010.

[8] H. van Tilborg and M. Blaum, "On Error-Correcting Balanced Codes", *IEEE Trans. Inform. Theory*, vol. 35, no. 5, pp. 1091-1095, September 1989.

[9] J.H. Weber and K.A. Schouhamer Immink, "Knuth's Balanced Code Revisited", *IEEE Trans. Inform. Theory*, vol. 56, no. 4, pp. 1673-1679, April 2010.

[10] J.H. Weber, K.A. Schouhamer Immink, and H.C. Ferreira, "Error-Correcting Balanced Knuth Codes", submitted to *IEEE Trans. Inform. Theory*, January 2011.