Knuth's Balancing of Codewords Revisited

Jos H. Weber
TU Delft, IRCTR/CWPC,
Mekelweg 4, 2628 CD Delft The Netherlands
Email: J.H.Weber@ewi.tudelft.nl

Kees A. Schouhamer Immink
Turing Machines Inc.
Willemskade 15b-d, 3016 DK Rotterdam The Netherlands
Email: immink@turing-machines.com

***Abstract*— In 1986, Don Knuth published a very simple algorithm for constructing sets of bipolar codewords with equal numbers of '1's and '-1's, called balanced codes. Knuth's algorithm is, since look-up tables are absent, well suited for use with large codewords. The redundancy of Knuth's balanced codes is a factor of two larger than that of a code comprising the full set of balanced codewords. In our paper we will present results of our attempts to improve the performance of Knuth's balanced codes.**

Key words: magnetic recording, optical recording, channel capacity, constrained code, balanced code.

I. Introduction

Sets of bipolar codewords that have equal numbers of '1's and '-1's are usually called *balanced codes*. Such codes have found application in cable transmission, optical and magnetic recording. A survey of properties and methods for constructing balanced codes can be found in [1]. A simple encoding technique for generating balanced codewords, which is capable of handling (very) large blocks was described by Knuth [2] in 1986.

Knuth's algorithm is extremely simple. An m-bit user word, m even, consisting of bipolar symbols valued ∓ 1 is forwarded to the encoder. The encoder inverts the first k bits of the user word, where k is chosen in such a way that the modified word has equal numbers of '1's and '-1's. Knuth showed that such an index k can always be found. The index k is represented by a (preferably) balanced word $\mathbf{u}$ of length p. The p-bit prefix word followed by the modified m-bit user 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 k bits received. Both encoder and decoder do not require look-up tables, and Knuth's algorithm is therefore very attractive for constructing long balanced codewords. Modifications of the generic scheme are discussed in Knuth [2], Alon *et al.* [3], Al-Bassam & Bose [4], and Tallini, Capocelli & Bose [5].

Knuth showed that in his best construction p is roughly equal to $\log_2 m + 1/2 \log_2 \log_2 m$ so that the redundancy of Knuth's construction is [2]

$$\frac{1}{m}\log_2 m, \quad m >> 1. \tag{1}$$

The cardinality of a full set of balanced codewords of length m equals

$$\binom{m}{\frac{m}{2}} \approx \frac{2^m}{\sqrt{\frac{m\pi}{2}}}, \quad m >> 1,$$

where the approximation of the central binomial coefficient is due to Stirling. Then the redundancy of a full set of balanced codewords is

$$\frac{1}{2m}\log_2 m, \quad m >> 1. \tag{2}$$

We conclude that the redundancy of a balanced code generated by Knuth's algorithm falls a factor of two short with respect to a code that uses 'full' balanced code sets. Clearly, the loss in redundancy is the price one has to pay for a simple construction. There are two features of Knuth's construction that could help to explain the difference in performance, and they offer opportunities for code improvement.

The first feature that may offer a possibility of improving the code's performance stems from the fact that Knuth's algorithm is greedy as it takes the very first opportunity for balancing the codeword [1], that is, in Knuth's basic scheme, the first, i.e. the smallest index k where balance is reached is selected. In case there is more than one position where balance can be achieved, the encoder will thus favor smaller values of the position index. As a result, we may expect that smaller values of the index k are more probable than larger ones. Then, if the index distribution is non-uniform, we may conclude that the average length of the prefix required to transmit the position information is less than $\log_2 m$. A practical embodiment of a scheme that takes advantage of this feature is characterized by the fact that the length of the prefix word is not fixed, but user data dependent. The prefix assigned to a position with a smaller, more probable, index has a smaller length than a prefix assigned to a position with a larger index.

Secondly, it has been shown by Knuth that there is always a position where balance can be reached. It can be verified that there is, for some user words, more than one suitable position where balance of the word can be realized. It will be shown later that the number of positions where words can be balanced lies between 1 and $m/2$. This freedom offers a possibility to improve the redundancy of Knuth's basic construction. An enhanced Knuth's algorithm may transmit auxiliary data by using the freedom of selecting from the balancing positions possible. Assume there are v positions, $1 \leq v \leq m/2$, where the encoder can balance the user word, then the encoder can convey an additional $\log_2 v$ bits. The number v depends on the user word, and therefore the amount of auxiliary data that can be transmitted is user data dependent.

We start, in Section II, with a survey of known properties of Knuth's coding method. Thereafter, we will compute the

978-1-4244-2571-6/08/$25.00 ©2008 IEEE

distribution of the transmitted index in Section III. Given the distribution of the index, we will compute the entropy of the index, and evaluate the performance of a suitably modified scheme. In Section IV, we will compute the amount of additional data that can be conveyed in a modification of Knuth's basic scheme. Section V concludes this article.

II. Knuth's basic scheme

Knuth's balancing algorithm is based on the idea that there is a simple translation between the set of all m-bit bipolar user words, m even, and the set of all $(m+p)$-bit codewords. The translation is achieved by selecting a bit position k within the m-bit word that defines two segments, each having the same, but opposite, disparity. A zero-disparity or balanced block is now generated by the inversion of the first k bits (or the last $m-k$ bits). The position digit k is encoded in the p-bit prefix. The rate of the code is simply $m/(m+p)$.

The proof that there is at least one position, k, where balance in any even length user word can be achieved is due to Knuth. Let the user word be $\mathbf{w} = (w_1, \ldots, w_m)$, $w_i \in \{-1, 1\}$, and let $d(\mathbf{w})$ be the sum, or disparity, of the user symbols, or

$$d(\mathbf{w}) = \sum_{i=1}^{m} w_i. \tag{3}$$

Let $d_k(\mathbf{w})$ be the running digital sum of the first k, $k \le m$, bits of $\mathbf{w}$, or

$$d_k(\mathbf{w}) = \sum_{i=1}^{k} w_i, \tag{4}$$

and let $\mathbf{w}^{(k)}$ be the word $\mathbf{w}$ with its first k bits inverted. For example, let

$$\mathbf{w} = (-1, 1, 1, 1, -1, 1, -1, 1, 1, -1),$$

then we have $d(\mathbf{w}) = 2$ and $\mathbf{w}^{(4)}$ = (1, -1, -1, -1, -1, 1, -1, 1, 1, -1). We let $\sigma_k(\mathbf{w})$ stand for $d(\mathbf{w}^{(k)})$, then the quantity $\sigma_k(\mathbf{w})$ is

$$\begin{aligned} \sigma_k(\mathbf{w}) &= -\sum_{i=1}^{k} w_i + \sum_{i=k+1}^{m} w_i \\ &= -2\sum_{i=1}^{k} w_i + d(\mathbf{w}). \end{aligned} \tag{5}$$

It is immediate that $\sigma_0(\mathbf{w}) = d(\mathbf{w})$, (no symbols inverted) and $\sigma_m(\mathbf{w}) = -d(\mathbf{w})$ (all m symbols inverted). We may, as $\sigma_{k+1}(w) = \sigma_k(w) \mp 2$, conclude that every word $\mathbf{w}$, m even, can be associated with at least one position k for which $\sigma_k(\mathbf{w}) = 0$, or $\mathbf{w}^{(k)}$ is balanced. This concludes the proof.

The value of k is encoded in a (preferably) balanced word $\mathbf{u}$ of length p, p even. The maximum codeword length of $\mathbf{w}$ is, since the prefix has an equal number of '1's and '-1's, governed by

$$m \le \binom{p}{p/2}. \tag{6}$$

In this article, we follow Knuth's generic format, where $1 \le k \le m$. Note that in a slightly different format, we may opt for $0 \le k \le m$, where the encoder has the option to invert or not to invert the codeword in case the user word is balanced. For small values of m, this will lead to slightly different results, though for very large values of m, the differences between the two formats are small. Knuth described some variations on the general framework. For example, if m and p are both odd, we can use a similar construction. The redundancy of Knuth's most efficient construction is

$$\frac{1}{m}\log_2 m, \quad m >> 1.$$

III. Distribution of the transmitted index

The basic Knuth algorithm, as described above, progressively scans the user word till it finds the *first* suitable position, k, where the word can be balanced. In case there is more than one position where balance can be obtained, it is expected that the encoder will favor smaller values of the position index. Then the distribution of the index k is not uniform, and, thus, the entropy of the index is less than $\log_2 m$, which opens the door for a more efficient scheme. A practical embodiment of a more efficient scheme would imply that the prefix assigned to a smaller index has a smaller length than a prefix assigned to a larger index. We will compute the entropy of the index sent by the basic Knuth encoder, and in order to do so we first compute the probability distribution of the transmitted index. In our analysis it is assumed that all information words are equiprobable and independent. Let $Pr_1(k)$ denote the probability that the transmitted index equals k, $1 \le k \le m$.

Theorem 1: The distribution of the transmitted index k, $Pr_1(k)$, $1 \le k \le m$ is given by $(1 \le j \le m/2)$

$$\begin{aligned} Pr_1(2j) &= Pr_1(2j-1) \\ &= \frac{m-2j+1}{m2^{m-2}}\binom{2(j-1)}{j-1}\binom{2(m/2-j)}{m/2-j}. \end{aligned}$$

Proof. This result can be shown using *On Line Encyclopedia of Integer Sequences* A33820. ■

Invoking Stirling's approximation, we have

$$\begin{aligned} Pr_1(2j) &= Pr_1(2j-1) \\ &\approx \frac{2}{\pi m}\frac{m-2j+1}{\sqrt{(2j-2)(m-2j)}}, 1 < j < m/2. \end{aligned}$$

For $j = 1$, we have $Pr_1(1) = Pr_1(2) \approx \frac{m-1}{m}\sqrt{\frac{2}{\pi m}}$, and for $j = m/2$, we have $Pr_1(m) = Pr_1(m-1) \approx \frac{1}{m}\sqrt{\frac{2}{\pi m}}$. Figure 1 shows two examples of the distribution, $Pr_1(j)$, for $m = 64$ and $m = 256$. The entropy of the transmitted index, denoted by $H_p(m)$, is

$$H_p(m) = -\sum_{k=1}^{m} Pr_1(k)\log_2 Pr_1(k). \tag{7}$$

Given the distribution, it is now straightforward to compute the entropy, $H_p(m)$, of the index. Figure 2 shows a few results of computations. The diagram shows that $H_p(m)$ is only slightly

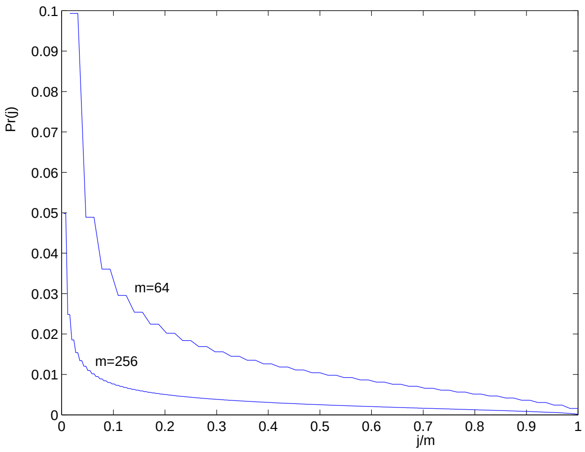

Fig. 1. Distribution $Pr_1(j)$ of the (normalized) transmitted index j/m for $m = 64$ and $m = 256$.

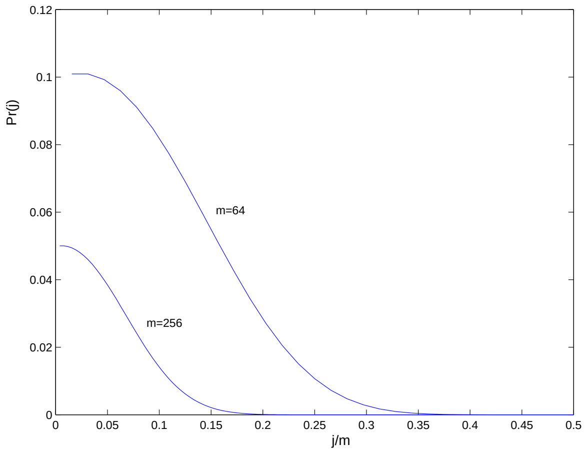

Fig. 3. Distribution $Pr_2(j)$ of the (normalized) number, j/m, of possible balancing positions for $m = 64$ and $m = 256$.

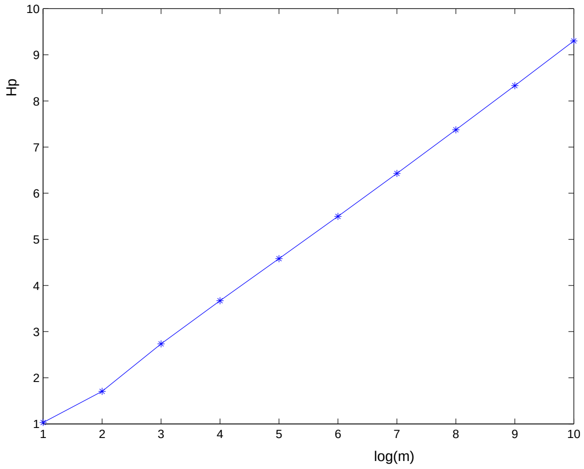

Fig. 2. Entropy $H_p(m)$ versus $\log_2(m)$.

less than $\log_2 m$, and we conclude that the above proposed modification of Knuth's scheme using a variable length prefix can offer only a small improvement in redundancy within the range of codeword length investigated. We conclude that, within the range of codeword length investigated, the proposed variable prefix-length scheme cannot bridge the factor of two in redundancy between the basic Knuth scheme and that of full set balanced codes.

IV. Encoding auxiliary data

There is at least one position and there at most $m/2$ positions within an m-bit word, m even, where a word can be balanced. The 'at least' one position, which makes Knuth's algorithm possible, was proved by Knuth (see above). The 'at most' bound will be shown in the next Theorem.

Theorem 2: There at most $m/2$ positions within an m-bit word, m even, where a word can be balanced.

Proof. Let k denote the position where balance can be made. Then, at the neighboring positions $k+1$ or $k-1$ such a balance cannot be made, so that we conclude that the number of positions where balance can be made is less or equal to $m/2$. ■

Note that the indices of a word with $m/2$ balance positions are either all even or all odd. It can easily be verified that there are three groups of words that can be balanced at $m/2$ positions. Namely

- a) the $2^{m/2}$ words consisting of the cascade of the $m/2$ di-bits (+1,-1) or (-1,+1),
- b) the $2^{m/2-1}$ words beginning with a +1 followed by $m/2-1$ di-bits (+1,-1) or (-1,+1), followed by a +1, and
- c) the inverted words of case b).

Since, on average, the encoder has the degree of freedom of selecting from more than one balance position, it offers the encoder the possibility to transmit auxiliary data. Assume there are v positions, $1 \le v \le m/2$, where the encoder can balance the user word, then the encoder can convey an additional $\log_2 v$ bits. The number v depends on the user word at hand, and therefore the amount of auxiliary data that can be transmitted is user data dependent.

Let $Pr_2(v)$ denote the probability that the encoder may choose between v, $1 \le v \le m/2$, possible positions, where balancing is possible.

Theorem 3: The distribution of the number of positions, where an m-bit word, m even, can be balanced is given by

$$Pr_2(v) = 2^{v+1-m}\binom{m-1-v}{m/2-v}, 1 \le v \le m/2. \quad (8)$$

Proof. See Appendix. Theorem 3 follows from Lemma 3 and the fact that there are 2^m sequences of length m, which are assumed to be equally probable. ■

Figure 3 shows two examples of the distribution, namely for $m = 64$ and $m = 256$. The average amount of information, $H_a(m)$, that can be conveyed via the choice in the position

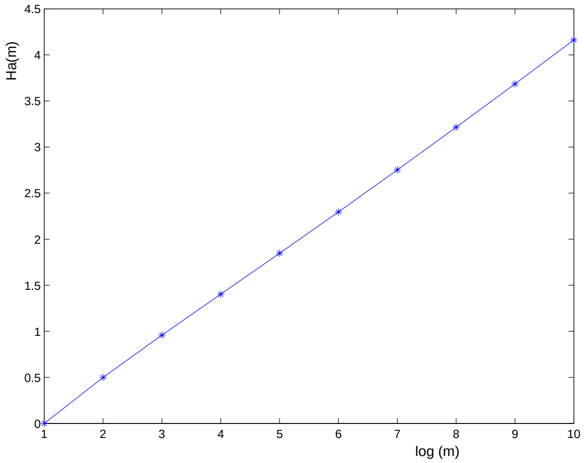

Fig. 4. The average amount of information, $H_a(m)$, that can be conveyed via the choice in the index as a function of $\log_2(m)$.

data is

$$H_a(m) = \sum_{v=1}^{m/2} Pr_2(v) \log_2 v. \qquad (9)$$

Results of computations are shown in Figure 4. We can recursively compute $Pr_2(v)$ by invoking

$$Pr_2(v) = \frac{m-2v+2}{m-v} Pr_2(v-1), \quad 2 \le v \le m/2.$$

For large m and $v << m$, we have

$$Pr_2(v) \approx Pr_2(v-1)(1-\frac{v-2}{m}),$$

where $Pr_2(1) \approx \frac{1}{\sqrt{\pi(m-2)/2}}$. We approximate

$$(1-\frac{1}{m})(1-\frac{2}{m})(1-\frac{3}{m})\dots(1-\frac{v-2}{m}) \approx e^{\frac{-(v-2)^2}{2m}},$$

so that

$$Pr_2(v) \approx \frac{1}{\sqrt{\pi(m-2)/2}} e^{\frac{-(v-2)^2}{2m}}, \quad 2 \le v \le m/2.$$

Now, for large m, we can approximate $H_a(m)$ by

$$\begin{aligned} H_a(m) &\approx \frac{1}{\sqrt{\pi(m-2)/2}} \int_{v=0}^{\infty} e^{\frac{-v^2}{2m}} \log_2 v dv && (10) \\ &\approx \frac{1}{2}\log_2 m - \frac{1}{2}(1+\frac{\gamma}{\ln 2}) && (11) \\ &\approx \frac{1}{2}\log_2 m - 0.916, && (12) \end{aligned}$$

where $\gamma \approx 0.57721$ is Euler's constant. We conclude that the average amount of information that can be conveyed by exploiting the choice of index compensates for the loss in rate between codes based on Knuth's algorithm and codes based on full balanced codeword sets.

V. Conclusions

We have investigated some characteristics and possible improvements of Knuth's algorithm for constructing bipolar codewords with equal numbers of '1's and '-1's. An $(m+p)$-bit codeword is obtained after a small modification of the m-bit user word plus appending a, fixed-length, p-bit prefix. The p-bit prefix represents the position index within the codeword, where the modification has been made.

We have derived the distribution of the index (assuming equiprobable user words), and have computed the entropy of the transmitted index. Our computations show that a modification of Knuth's generic scheme using a variable length prefix of the position index will only offer a small improvement in redundancy.

The transmitter can, in general, choose from a plurality of indices, so that the transmitter can transmit additional information. The number of possible indices depends on the given user word, so that the amount of extra information that can be transmitted is data dependent. We have derived the distribution of the number of positions where a word can be balanced. We have computed the average information that can be conveyed by using the freedom of choosing from multiple indices. The average information rate can, for large user word length, m, be approximated by $\frac{1}{2}\log_2 m - 0.916$. This compensates for the loss in code rate between codes based on Knuth's algorithm and codes based on full balanced codeword sets.

VI. Appendix

In this appendix we give a combinatorial proof of Theorem 3. We also refer the reader to *On Line Encyclopedia of Integer Sequences* A112326. Let $\mathcal{A}_v(m)$ denote the set of bipolar sequences of even length m which can be balanced in v positions $(1 \le v \le m/2)$. Define $A_v(m) = |\mathcal{A}_v(m)|\ \forall m, v$. We will derive an explicit expression for $A_v(m)$ (in Lemma 3), from which Theorem 3 immediately follows. Let $\mathcal{B}(m)$ denote the set of all balanced sequences of length m without internal balancing positions, i.e., there are no balancing positions j with $0 < j < m$. Define $B(m) = |\mathcal{B}(m)|\ \forall m$. Any sequence $\mathbf{a} \in \mathcal{A}_v(m)$ with balancing positions $\{j_1, j_2, \dots, j_v\}$ can be uniquely decomposed as $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_v, \mathbf{a}_{v+1})$, where $\mathbf{a}_i$ is of length $j_i - j_{i-1}$, with $j_0 = 0$ and $j_{v+1} = m$. Note that $\mathbf{a}_i$ is in $\mathcal{B}(j_i - j_{i-1})$ for all $2 \le i \le v$ and that $(\mathbf{a}_1, \mathbf{a}_{v+1})$ is in $\mathcal{A}_1(m - j_v + j_1)$. From these observations, we can easily derive the recursive relation

$$A_v(m) = \sum_{i=1}^{m/2-v+1} A_{v-1}(m-2i)B(2i) \qquad (13)$$

for all $2 \le v \le m/2$. Further, we have, for all m, the trivial equality

$$\sum_{v=1}^{m/2} A_v(m) = 2^m. \qquad (14)$$

A *Dyck word* of length $2n \ge 0$ is a balanced bipolar sequence of length $2n$ such that no initial segment has more '1's than

'-1's [6], or in other words, $\mathbf{w}$ is a Dyck word if the running digital sum $d_k(\mathbf{w}) = \sum_{i=1}^{k} w_i \le 0$ for all $1 \le k \le 2n$. The number of Dyck words of length $2n$ is equal to

$$C_n = \frac{1}{n+1}\left(\begin{array}{c} 2n \\ n \end{array}\right), \tag{15}$$

which is the n-th Catalan number [6]. For example, 0011, and 0101 are the $C_2 = 2$ Dyck words of length 4, and 000111, 001011, 001101, 010011, and 010101 are the $C_3 = 5$ Dyck words of length 6, where for clerical convenience we have written '0' instead of '-1'. Note that a sequence $\mathbf{b}$ is in $\mathcal{B}(m)$ if and only if it has (the inverse of) the format $(0, \mathbf{d}, 1)$, where d is a Dyck word of length $m-2$. Hence, for all m,

$$B(m) = 2C_{(m-2)/2} = \frac{4}{m}\left(\begin{array}{c} m-2 \\ m/2-1 \end{array}\right). \tag{16}$$

For example, $B(6) = |\mathcal{B}(6)| = |\{000111,\ 001011,\ 111000,\ 110100\}| = 4$, which is indeed the result provided by (16).

Lemma 1: For all t and u satisfying $1 \le u \le t$, it holds that

$$\sum_{i=1}^{t-u+1}\left(\begin{array}{c} 2t-2i-u+1 \\ t-i-u+1 \end{array}\right)\frac{1}{i}\left(\begin{array}{c} 2i-2 \\ i-1 \end{array}\right) = \left(\begin{array}{c} 2t-u \\ t-u \end{array}\right). \tag{17}$$

Proof. Any bipolar sequence of length $2t-u$ containing t 'ones' can be uniquely written as $(\mathbf{x}, 1, \mathbf{y})$, where $\mathbf{x}$ is a Dyck word of length $2i-2$, with $i \in \{1, 2, \ldots, t-u+1\}$, and $\mathbf{y}$ is a bipolar sequence of length $2t-u-2i+1$ containing $t-i$ 1's. Using (15) for Dyck word enumeration, a simple counting argument gives the stated result. ■

Lemma 2: For all $t \ge 0$, it holds that

$$\sum_{i=0}^{t} 2^i \left(\begin{array}{c} 2t-i \\ t-i \end{array}\right) = 2^{2t}. \tag{18}$$

Proof. Any bipolar sequence of length $2t$ having more than t 1's can be uniquely written as $(\mathbf{x}, 1, \mathbf{y})$, where $\mathbf{x}$ is of length $i-1$, with $i \in \{1, 2, \ldots, t\}$, and $\mathbf{y}$ is of length $2t-i$ and has t 1's. Any bipolar sequence of length $2t$ containing less than t 1's can be uniquely written as $(\mathbf{x}, -1, \mathbf{y})$, where $\mathbf{x}$ is of length $i-1$, with $i \in \{1, 2, \ldots, t\}$, and $\mathbf{y}$ is of length $2t-i$ and has $t-i$ 1's. Hence,

$$\begin{aligned} 2^{2t} &= \\ &= \sum_{i=1}^{t} 2^{i-1}\left(\begin{array}{c} 2t-i \\ t \end{array}\right) + \left(\begin{array}{c} 2t \\ t \end{array}\right) + \sum_{i=1}^{t} 2^{i-1}\left(\begin{array}{c} 2t-i \\ t-i \end{array}\right) \\ &= \sum_{i=0}^{t} 2^{i}\left(\begin{array}{c} 2t-i \\ t-i \end{array}\right), \end{aligned} \tag{19}$$

which concludes the proof. ■

Lemma 3: For all $1 \le v \le m/2$, it holds that

$$A_v(m) = 2^{v+1}\left(\begin{array}{c} m-1-v \\ m/2-v \end{array}\right) \tag{20}$$

Proof. Assuming that the statement holds for all $m \le m^*$, we will show that it also holds for $m = m^* + 2$. For all $2 \le v \le (m^*+2)/2$, we have

$$\begin{aligned} A_v(m^*+2) &= \\ &= \sum_{i=1}^{m^*/2-v+2} A_{v-1}(m^*+2-2i)B(2i) \\ &= \sum_{i=1}^{m^*/2-v+2} 2^v \left(\begin{array}{c} m^*+2-2i-v \\ m^*/2+2-i-v \end{array}\right)\frac{2}{i}\left(\begin{array}{c} 2i-2 \\ i-1 \end{array}\right) \\ &= 2^{v+1}\left(\begin{array}{c} m^*+1-v \\ m^*/2+1-v \end{array}\right), \end{aligned} \tag{21}$$

where the first equality follows from (13), the second from (20) and (16), and the third from Lemma 1 (with $t = m^*/2$ and $u = v-1$). Further, we have

$$\begin{aligned} A_1(m^*+2) &= \\ &= 2^{m^*+2} - \sum_{v=2}^{m^*/2+1} A_v(m^*+2) \\ &= 4\left(2^{m^*} - \sum_{v=2}^{m^*/2+1} 2^{v-1}\left(\begin{array}{c} m^*+1-v \\ m^*/2+1-v \end{array}\right)\right) \\ &= 4\left(\begin{array}{c} m^* \\ m^*/2 \end{array}\right), \end{aligned} \tag{22}$$

where the first equality follows from (14) (with $m = m^*+2$), the second from (21), and the third from Lemma 2 (with $t = m^*/2$). Hence, if the statement in the lemma holds for all $m \le m^*$, then it holds for $m = m^*+2$ as well. Since (14) gives that $A_1(2) = 4$, (20) holds for $m = 2$, and the lemma follows by induction on m. ■

REFERENCES

[1] K.A.S. Immink, *Codes for Mass Data Storage Systems,* Second Edition, ISBN 90-74249-27-2, Shannon Foundation Publishers, Eindhoven, Netherlands, 2004.

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

[3] N. Alon, E.E. Bergmann, D. Coppersmith, and A.M. Odlyzko, 'Balancing Sets of Vectors', *IEEE Trans. Inform. Theory,* vol. IT-34, no. 1, pp. 128-130, Jan. 1988.

[4] S. Al-Bassam and B. Bose, 'On Balanced Codes', *IEEE Trans. Inform. Theory,* vol. IT-36, no. 2, pp. 406-408, March 1990.

[5] L.G. Tallini, R.M. Capocelli, and B. Bose, 'Design of Some New Balanced Codes', *IEEE Trans. Inform. Theory,* vol. IT-42, no. 3, pp. 790-802, May 1996.

[6] R.P. Stanley, *Enumerative Combinatorics,* Vol. 2, Cambridge University Press, 1999.