Efficient Balancing of q-ary Sequences with Parallel Decoding

Theo G. Swart
Dept. of Electrical and Electronic Engineering Science,
University of Johannesburg,
P.O. Box 524, Auckland Park, 2006, South Africa
Email: tgswart@uj.ac.za

Jos H. Weber
Delft University of Technology, IRCTR/CWPC,
Mekelweg 4,
2628 CD Delft The Netherlands
Email: j.h.weber@ewi.tudelft.nl

***Abstract*—Balancing of q-ary sequences, using a generalization of Knuth's efficient parallel balancing scheme, is considered. It is shown that the new general scheme is as simple as the original binary scheme, which lends itself to parallel decoding of the balanced sequences.**

I. Introduction

Balanced codes have found applications in various systems, such as optical and magnetic recording, cable transmissions, detection of unidirectional errors and reducing noise in VLSI systems. In many of these systems it is advantageous to have an efficient encoding and decoding method. Fast, parallel decoding is also an advantage in optical systems where very high communication speeds are encountered.

Knuth [1] presented such efficient methods for binary sequences. It was shown that any binary sequence can be balanced by complementing the bits up to a certain point. In the parallel method, the balanced sequence can then be transmitted, as well as information regarding the position of the balancing point. Based on this information, the receiver can recover the original sequence. A serial method was also presented, that used this balancing in a different manner. Since then various results based on this idea and others have been presented for the binary case [2]–[6].

In the design of q-ary immutable codes [7] it was found that the maximum size code was obtained when balanced sequences were used. All the q-ary sequences are partitioned into disjoint chains and then using two functions, that must satisfy certain properties, any q-ary sequence can be encoded into a sequence that is balanced. Knuth's idea has also been extended to balancing of q-ary sequences in [8]. Sequences that are close to being balanced are encoded with single or double maps. These maps make use of a generalization of Knuth's serial method. The other sequences are compressed with a uniquely decodable variable length code [5] and balanced using the saved space.

Work closely related to q-ary balancing includes balancing codes over the mth roots of unity [9], [10] and balancing codes that are invariant under symbol permutation [11].

In this paper we will present a generalization of Knuth's parallel method, resulting in a q-ary scheme that is much simpler in decoding than previous ones.

A q-ary sequence $\mathbf{x} = x_0 x_1 \ldots x_{n-1}$, $x_i \in \{0, 1, \ldots, q-1\}$, of length n is balanced if

$$\sum_{i=0}^{n-1} x_i = n(q-1)/2.$$

Let β denote this balancing value. This holds for all q and n, except when q is even and n is odd. For the rest of the paper we will not consider sequences with q even and n odd, so that $n(q-1)/2$ is an integer.

An alternative view is to consider a q-ary sequence $\mathbf{x} = x_0 x_1 \ldots x_{n-1}$ of length n with positive and negative components,

$$x_i \in \{-(q-1)/2, \ldots, -2, -1, 0, +1, +2, \ldots, +(q-1)/2\},$$

if q is odd and

$$x_i \in \{-(q-1), \ldots, -3, -1, +1, +3, \ldots, +(q-1)\},$$

if q is even. Then balancing is achieved whenever the components sum to zero,

$$\sum_{i=0}^{n-1} x_i = 0.$$

The conversion between the two representations is easily accomplished using a mapping

$$\{0, 1, \ldots, q-1\} \rightarrow \{-(q-1)/2, \ldots, -2, -1, 0, +1, +2, \ldots, +(q-1)/2\},$$

for q odd and

$$\{0, 1, \ldots, q-1\} \rightarrow \{-(q-1), \ldots, -3, -1, +1, +3, \ldots, +(q-1)\},$$

for q even. For simplicity in presentation, we will only use the former representation in this paper.

In Section II we generalize Knuth's balancing of binary sequences to balancing of q-ary sequences, in Section III we investigate the redundancy and complexity of the new scheme as well as comparing it to previous ones, and finally we conclude with Section IV.

978-1-4244-4313-0/09/$25.00 ©2009 IEEE

II. Balancing of q-ary sequences

Knuth's method was based on the fact that any binary sequence can be balanced by inverting the bits up to a certain point, or equivalently, adding a binary sequence modulo two. The sequences being added to the original sequence will be called *balancing sequences*. We illustrate this by an example.

Example 1 Consider a binary sequence 11101011, with the sum of components equal to 6. If the first two bits are inverted, we have the sequence 00101011, which is balanced with the sum of components equal to $\beta = 4$. Balancing can also be achieved after inverting four bits (00011011) and six bits (00010111). This is equivalent to adding a sequence modulo two to the original sequence, as follows

$$11101011 \oplus 11000000 = 00101011,$$
$$11101011 \oplus 11110000 = 00011011,$$
$$11101011 \oplus 11111100 = 00010111.$$

Using this method, the receiver only needs to know up to which position the bits were inverted.

Balancing can also be achieved by inverting the bits from the opposite side, as in

$$11101011 \oplus 00000011 = 11101000,$$
$$11101011 \oplus 00001111 = 11100100,$$
$$11101011 \oplus 00111111 = 11010100.$$

For reasons that will become clear shortly, we will use 22222211, 22221111 and 22111111 instead for these latter balancing sequences. This does not affect the final result since $0 \equiv 2 \pmod 2$. □

By generalizing Knuth's method, we will balance a q-ary sequence by adding modulo q an appropriate $(q+1)$-ary balancing sequence, as shown in the next example.

Example 2 Consider a 4-ary sequence 02333132, with the sum of components equal to 17. By adding the sequence 22222221, we can obtain a balanced sequence 20111313 with the sum of components equal to $\beta = 12$. The following sequences can be obtained that balance the sequence,

$$02333132 \oplus_4 22222221 = 20111313,$$
$$02333132 \oplus_4 33322222 = 31211310,$$
$$02333132 \oplus_4 33333332 = 31222020,$$
$$02333132 \oplus_4 44433333 = 02322021,$$

where $\oplus_4$ is used to denote modulo 4 addition. □

We will now formally define the balancing sequence.

Definition 1 A $(q+1)$-ary balancing sequence of length n, denoted by $\mathbf{b}(s,p)$ where $0 \le s \le q-1$ and $0 \le p \le n-1$, is of the form $\mathbf{b}(s,p) = b_0 b_1 \dots b_{n-1}$ with

$$b_i = \begin{cases} s, & i \ge p, \\ s+1, & \text{otherwise.} \end{cases}$$ □

Using the balancing sequences from the previous example, we have

$$\mathbf{b}(1,7) = 22222221, \quad \mathbf{b}(2,3) = 33322222,$$
$$\mathbf{b}(2,7) = 33333332, \quad \mathbf{b}(3,3) = 44433333.$$

Remark 1 A $(q+1)$-ary balancing sequence is used for simplicity in presentation, but for ease of understanding one can see it as two sequences, consisting of a q-ary sequence (the all-s sequence) and a "binary" sequence (indicating the position). Then,

$$\mathbf{b}(s,p) = sss\dots s \oplus_q \overbrace{1\dots1}^{p} 0\dots0.$$ □

Let $\mathbf{y}$ be the sequence after a balancing sequence is added, i.e. $\mathbf{y} = \mathbf{x} \oplus_q \mathbf{b}(s,p)$, and let $\sigma = \sum_{i=0}^{n-1} y_i$. Note there are qn possible balancing sequences. To see how the balancing sequences affect σ, let z denote the z-th balancing sequence, with

$$z = sn + p, \quad 0 \le z \le qn - 1$$

and let $\sigma(z)$ denote the sum of components when adding the z-th balancing sequence to the original sequence. Then, the original sequence's sum of components is given by $\sigma(0)$ when the all-zeros sequence is added. Since the next sequence after $qqq\dots q(q-1)$ would be $qqq\dots qq$, and $000\dots00 \equiv qqq\dots qq$ under modulo q addition, we have that $\sigma(0) = \sigma(qn)$. Thus, for $z \ge qn$ the balancing sequences would start repeating. By abuse of notation, $\mathbf{b}(z)$ will also be used to denote the z-th balancing sequence and $\mathbf{y}(z)$ will denote the z-th sequence obtained after adding $\mathbf{b}(z)$, i.e. $\mathbf{y}(z) = \mathbf{x} \oplus_q \mathbf{b}(z)$.

We use the following example to illustrate how sequences are balanced using the balancing sequences.

Example 3 Consider the binary sequence 11101011 from Example 1, with $\sigma(0) = 6$. Fig. 1 shows the $\sigma(z)$-values graphed against z, the dashed line indicates when balancing is achieved with $\beta = 4$. As was seen in that example, there are six balancing sequences that achieve balancing. □

For the binary case, we know from [1] that the minimum and maximum values of $\sigma(z)$ will always be such that $\min\{\sigma(z)\} \le \beta \le \max\{\sigma(z)\}$. The increase and decrease in σ in the graph will always be one, and therefore it must pass through β at some stage. A graph or path of this nature is called a *random walk*, and this formed the basis of Knuth's proof. For q-ary balancing in [8], Tallini and Vaccaro construct single or double maps in such a way that random walks are

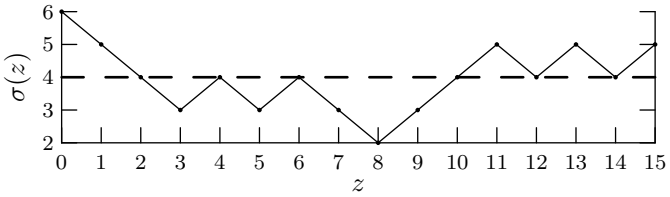

Fig. 1. (1,1)-random walk for 11101011

also achieved, with σ changing by -1, 0 or $+1$ for each step, thereby ensuring that $\sigma(z)$ will pass through a certain value. Our approach does not need these single or double maps, and is as simple as Knuth's method.

The new scheme also results in random walks, but the value of σ does not change by ± 1. To generalize, define an (γ, τ)-random walk as a path with increases of γ and decreases of τ. The previous random walks would then be considered as $(1, 1)$-random walks.

The next example is used to understand the similarities and differences between the binary and non-binary cases.

Example 4 Consider the 4-ary sequence 02333132 from Example 2, with $\sigma(0) = 17$ and Fig. 2 showing the $\sigma(z)$-values with the dashed line indicating balancing when $\beta = 12$.

As determined in that example, there are four possible balancing sequences. From the figure we see that increases are always in increments of one, while decreases are in increments of 3. □

The path always starts at $\sigma(0)$ and returns to that value at $\sigma(qn)$, with several increments and decrements in between. If we can prove that $\sigma(z)$ is a $(1, q-1)$-random walk and that $\min\{\sigma(z)\} \leq \beta \leq \max\{\sigma(z)\}$, then β must be reached at some stage by one of the upward segments of the walk (possibly by downward segments as well).

Lemma 1 *When adding* $\mathbf{b}(z)$ *to* $\mathbf{x}$*,* $\sigma(z)$ *forms a* $(1, q-1)$*-random walk for* $0 \leq z \leq qn - 1$. □

Proof: The difference between $\mathbf{b}(z)$ and $\mathbf{b}(z+1)$ is the symbol in position p, with $b_p(z+1) - b_p(z) = 1$. Thus, $\sigma(z+1) - \sigma(z)$ depends on the result when one is added to the value of $y_p(z)$, since

$$\begin{aligned} y_p(z+1) &= x_p \oplus_q b_p(z+1) \\ &= x_p \oplus_q b_p(z) \oplus_q 1 \\ &= y_p(z) \oplus_q 1. \end{aligned}$$

If $y_p(z) < q-1$ then $y_p(z+1) = y_p(z) + 1$ and $\sigma(z+1) - \sigma(z) = 1$. If $y_p(z) = q-1$ then $y_p(z+1) = y_p(z) - (q-1)$, since $q - 1 \oplus_q 1 = 0$ and $\sigma(z+1) - \sigma(z) = -(q-1)$.

Therefore, $\sigma(z)$ forms a $(1, q-1)$-random walk, with either increases of one or decreases of $q-1$, when z increases by one. ■

Calculating an exact minimum and maximum value for $\sigma(z)$ is difficult since it is sequence dependent, but in the next lemma we will show that a bound exists on these values for all sequences.

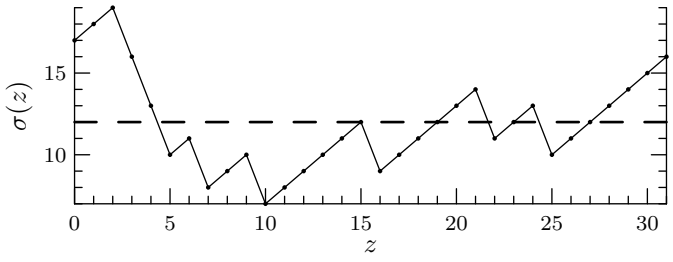

Fig. 2. (1,3)-random walk for 02333132

Lemma 2 *The* $(1, q-1)$*-random walk* $\sigma(z)$ *has* $\min\{\sigma(z)\} \leq \beta$ *and* $\max\{\sigma(z)\} \geq \beta$. □

Proof: Consider adding an all-s sequence, $\mathbf{b}(s, 0)$ to the original starting sequence $\mathbf{x}$ and calculate the real sum of

$$\sigma(\mathbf{x} \oplus_q \mathbf{b}(0,0)) + \sigma(\mathbf{x} \oplus_q \mathbf{b}(1,0)) + \cdots + \sigma(\mathbf{x} \oplus_q \mathbf{b}(q-1,0)).$$

This sum can be calculated by taking into account the symbols that can occur in each position in all these sequences. In any coordinate position, all the symbols $0, 1, \ldots, q-1$ will appear exactly once, and since

$$0 + 1 + \cdots + q - 1 = q(q-1)/2,$$

we find that

$$\sum_{s=0}^{q-1} \sigma(\mathbf{x} \oplus_q \mathbf{b}(s,0)) = nq(q-1)/2.$$

The average σ-value for all the q sequences is

$$\sigma_{\text{avg}} = n(q-1)/2 = \beta.$$

Hence, of the sequences $\mathbf{x} \oplus_q \mathbf{b}(s, 0)$, at least one has a σ-value of at least β and at least one has a σ-value of at most β. Even though the sequences $\mathbf{b}(s, 0)$ forms a subset of our balancing sequences, we can still conclude that $\min\{\sigma(z)\} \leq \beta \leq \max\{\sigma(z)\}$. ■

Theorem 1 *Any* q*-ary sequence of length* n *can be balanced by adding modulo* q *an appropriate balancing sequence* $\mathbf{b}(s, p)$. □

Proof: We know from Lemma 2 that $\min\{\sigma(z)\} \leq \beta \leq \max\{\sigma(z)\}$ and from Lemma 1 that $\sigma(z)$ is a $(1, q-1)$-random walk with unit increases. Therefore, it follows that $\sigma(z) = \beta$ for at least one z, i.e. at least one of the upward segments for $\sigma(z)$ must pass through β. Therefore balancing can always be achieved by at least one $\mathbf{b}(s, p)$. ■

For n even and $q = 2$ it is only necessary to use a binary balancing sequence, instead of a ternary balancing sequence, in which case it simplifies to Knuth's method. Note that in Fig. 1 the random walk goes through β six times when sequences up to $22\ldots21$ are used, compared to the three times when sequences up to $11\ldots10$ are used. This happens in the binary case because the second half of the random walk is an exact mirror image of the first half.

We also find that for certain n and q values a $(q+1)$-ary balancing sequence is not needed, and that a q-ary or even a $(q-1)$-ary balancing sequence can achieve balancing for all sequences, thereby lowering the redundancy slightly. However, this is only valid for low n values, such as $n = 2$ and $n = 4$ and since we are more interested in long sequences, these few exceptions will be ignored.

The next example shows why it is necessary to use balancing sequences up to $qqq\ldots q(q-1)$.

Example 5 Consider the 4-ary sequence 012223 with $\sigma(0) = 10$. Fig. 3 shows the (1,3)-random walk for $\sigma(z)$ of this sequence.

It can clearly be seen that balancing is only achieved when the balancing sequence is 444443. □

III. Redundancy and Complexity

Let $\mathcal{S}_q^n$ denote the cardinality for the full balanced set of q-ary sequences of length n. The cardinalities are listed in Table I. These values correspond to the central binomial coefficients (A001405), central trinomial coefficients (A002426), central quadrinomial coefficients (A005190) and central pentanomial coefficients (A005191) for sequences with $q = 2$, 3, 4 and 5 respectively, where the numbers indicate the sequences from [12].

From [13] and [7] we find that

$$\mathcal{S}_q^n = q^n \sqrt{\frac{6}{\pi n(q^2-1)}} \left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right)$$

Taking the logarithm, we obtain the approximation

$$\log_q(\mathcal{S}_q^n) \approx n - \frac{1}{2}\log_q n - \frac{1}{2}\log_q \frac{\pi}{6} - \frac{1}{2}\log_q(q^2-1).$$

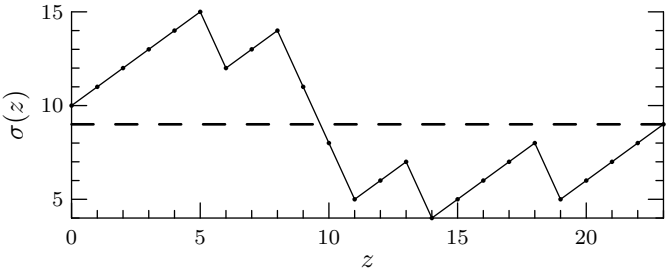

Fig. 3. (1,3)-random walk for 012223

TABLE I
Cardinalities of full balanced sets

	$q=2$	$q=3$	$q=4$	$q=5$
$n=4$	6	19	44	85
$n=5$		51		381
$n=6$	20	141	580	1751
$n=7$		393		8135
$n=8$	70	1107	8092	38165
$n=9$		3139		180325
$n=10$	252	8953	116304	856945
$n=11$		25653		4091495
$n=12$	924	73789	1703636	19611175
$n=13$		212941		94309099
$n=14$	3432	616227	25288120	454805755
$n=15$		1787607		2198649549
$n=16$	12870	5196627	379061020	10651488789
$n=17$		15134931		51698642405
$n=18$	48620	44152809	5724954544	251345549849
$n=19$		128996853		1223798004815
$n=20$	184756	377379369	86981744944	5966636799745

Hence, the redundancy of the full balanced set is roughly equal to

$$\frac{1}{2n}\log_q n, \quad n \gg 1. \tag{1}$$

In the new scheme, to transmit the balanced codeword successfully, it is necessary to transmit s and p for the receiver to recover the original codeword. The sequence used to communicate these values must also be balanced, thus we choose sequences from the full balanced set. The cardinality must be greater than or equal to qn to be able to transmit this information. A lookup table or enumerative encoding [14] can be used to encode and decode s and p, however the memory/complexity necessary to implement this is negligible in comparison to the memory/complexity when using a lookup table or enumerative encoding for the entire length n sequence.

Let k denote the length of the balanced sequence that will transmit the information regarding the chosen balancing sequence. We then require that

$$\mathcal{S}_q^k \geq qn,$$

and therefore, using a similar argument as above, we have

$$k - \frac{1}{2}\log_q k - \frac{1}{2}\log_q \frac{\pi}{6} - \frac{1}{2}\log_q(q^2-1) \geq \log_q n + 1.$$

Then, for very long sequences, the length of the added sequence is roughly

$$k \approx \log_q n, \quad n \gg 1,$$

making the redundancy in this case

$$\frac{1}{n}\log_q n, \quad n \gg 1.$$

The difference in redundancies between the full balanced set and the balanced set obtained using the new method is

$$\frac{1}{2n}\log_q n, \quad n \gg 1. \tag{2}$$

This is similar to the difference found in the redundancies for the binary case when using Knuth's method.

Weber and Immink [6] showed that this difference in redundancies can be overcome for the binary case, while still using Knuth's efficient method. Auxiliary information can be transmitted by the choice of balancing sequence. As was shown in Example 2, it is possible in some cases to have more than one balancing sequence. Additional information can then be transmitted by assigning information symbols to each balancing sequence, e.g.

$$\mathbf{b}(1,7) \to 0, \quad \mathbf{b}(2,3) \to 1,$$
$$\mathbf{b}(2,7) \to 2, \quad \mathbf{b}(3,3) \to 3.$$

Possible further work is to determine if the different balancing sequences for the q-ary case can also be used to transmit auxiliary information, thereby reducing (2).

In [7], codes with k redundant symbols can be constructed such that the length of information symbols is

$$n \leq \frac{q^k - 1}{q - 1}. \tag{3}$$

A similar, but slightly more complex code can be constructed such that

$$n \leq 2\frac{q^k - 1}{q - 1} - k. \tag{4}$$

In [8] codes are constructed with

$$n \leq \frac{1}{1 - 2\alpha}\frac{q^k - 1}{q - 1} - c_1(q, \alpha)k - c_2(q, \alpha),$$

where c_1 and c_2 are dependent on q and α, and $\alpha \in [0, \frac{1}{2})$. Balancing of sequences and compression are used in this construction. However, for comparison purposes we will focus on the balancing part of the construction, thereby not considering the compression method. Then the bound on the length of n simplifies to (4).

Reordering the equations and assuming the minimum number of redundant symbols, both (3) and (4), for very long sequences, are approximately

$$k \approx \log_q n, \quad n \gg 1,$$

with the redundancy approximately

$$\frac{1}{n}\log_q n, \quad n \gg 1.$$

Hence, the redundancies of the previous schemes and our new scheme are approximately the same for very long sequences. The redundancies of all these schemes are roughly a factor of two more than the redundancy of the full balanced set.

As in the binary case of Knuth, our decoding is also fast and efficient. Once the decoder has established s and p, $\mathbf{b}$ can be reconstructed and subtracted from $\mathbf{y}$ to recover $\mathbf{x}$. Since the entire sequence $\mathbf{b}$ is known, and no further calculations need to be done, all the symbol subtractions to obtain $\mathbf{x}$ can be done simultaneously, hence fast, parallel decoding of the balanced sequence is possible. In simple terms, decoding in previous schemes involved starting at a bit and changing it in a predetermined manner, until a certain weight was achieved. If this was not possible, the procedure was repeated on the following bit, and so forth. Thus far no parallel implementation has been proposed for this method of decoding in stages.

For implementation the previous schemes [7], [8] need $\mathcal{O}(qn \log_q n)$ digit operations for both encoding and decoding (as stated in [8]). In the worst case our new scheme needs $\mathcal{O}(qn \log_q n)$ digit operations for encoding (which is comparable to previous schemes) and only $\mathcal{O}(n)$ digit operations for decoding. Note that this does not take into account the complexity from using a lookup table or enumerative encoding.

Even though the constructions found in [7] and [8] have better redundancies compared to our new construction, our new approach allows for a simpler decoding. For the previous schemes, decoding is achieved by basically reversing the encoding steps. Our new approach only needs to subtract the already determined balancing sequence, which can be done in parallel, as stated previously.

IV. Conclusion

A simple construction was presented whereby any q-ary sequence can be balanced by adding a certain balancing sequence to it. The construction was shown to have simple and efficient encoding, as well as fast, parallel decoding.

The redundancies of the balanced codes obtained from this construction are just over a factor of two away from the redundancies of the full balanced sets. This can possibly be lowered by making use of auxiliary information, with only a slight increase in complexity.

References

[1] D. E. Knuth, "Efficient balanced codes," *IEEE Transactions on Information Theory*, vol. 32, no. 1, pp. 51–53, Jan. 1986.

[2] N. Alon, E. E. Bergmann, D. Coppersmith and A. M. Odlyzko, "Balancing sets of vectors," *IEEE Transactions on Information Theory*, vol. 34, no. 1, pp. 51–53, Jan. 1988.

[3] S. Al-Bassam and B. Bose, "Design of efficient balanced codes," *IEEE Transactions on Computers*, vol. 43, no. 3, pp. 362–365, Mar. 1994.

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

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

[6] J. H. Weber and K. A. S. Immink, "Knuth's balancing of codewords revisited," in *Proceedings IEEE International Symposium on Information Theory*, Toronto, Canada, July 6–11, 2008, pp. 1567-1571.

[7] R. M. Capocelli, L. Gargano and U. Vaccaro, "Efficient q-ary immutable codes," *Discrete Applied Mathematics*, vol. 33, pp. 25–41, 1991.

[8] L. G. Tallini and U. Vaccaro, "Efficient m-ary balanced codes", *Discrete Applied Mathematics*, vol. 92, pp. 17–56, 1999.

[9] A. Baliga and S. Boztaş, "Balancing sets of non-binary vectors", in *Proceedings of IEEE International Symposium on Information Theory*, Lausanna, Switzerland, Jun 30–Jul 5, 2002, p. 300.

[10] R. Mascella, L. G. Tallini, S. Al-Bassam and B. Bose, "On efficient balanced codes over the mth roots of unity," *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 2214–2217, May 2006.

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

[12] N. J. A. Sloane, "The on-line encyclopedia of integer sequences," 2005. [Online]. Available: http://www.research.att.com/~njas/sequences/

[13] Z. Star, "An asymptotic formula in the theory of compositions," *Aequationes Mathematicae*, vol. 13, pp. 279–284, 1975.

[14] T. Cover, "Enumerative source encoding," *IEEE Transactions on Information Theory*, vol. 19, no. 1, pp. 73–77, 1973.