<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Linear size optimal q-ary constant-weight codes and constant-composition codes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Chee, Yeow Meng; Dau, Son Hoang; Ling, Alan C. H.; Ling, San</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>2009</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10220/7637">http://hdl.handle.net/10220/7637</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>© 2009 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: <a href="http://dx.doi.org/10.1109/TIT.2009.2034814">http://dx.doi.org/10.1109/TIT.2009.2034814</a>.</td>
</tr>
</tbody>
</table>
Linear Size Optimal $q$-ary Constant-Weight Codes and Constant-Composition Codes
Yeow Meng Chee, Senior Member, IEEE, Son Hoang Dau, Alan C. H. Ling, and San Ling

Abstract—An optimal constant-composition or constant-weight code of weight $w$ has linear size if and only if its distance $d$ is at least $2w - 1$. When $d \geq 2w$, the determination of the exact size of such a constant-composition or constant-weight code is trivial, but the case of $d = 2w - 1$ has been solved previously only for binary and ternary constant-composition and constant-weight codes, and for some sporadic instances.

This paper provides a construction for quasicyclic optimal constant-composition and constant-weight codes of weight $w$ and distance $2w - 1$ based on a new generalization of difference triangle sets. As a result, the sizes of optimal constant-composition codes and optimal constant-weight codes of weight $w$ and distance $2w - 1$ are determined for all such codes of sufficiently large lengths. This solves an open problem of Etzion.

The sizes of optimal constant-composition codes of weight $w$ and distance $2w - 1$ are also determined for all $w \leq 6$, except in two cases.

Index Terms—constant-composition codes, constant-weight codes, difference triangle sets, generalized Steiner systems, Golomb rulers, quasicyclic codes

I. INTRODUCTION

There are two generalizations of binary constant-weight codes as we enlarge the alphabet beyond size two. These are the classes of constant-composition codes and $q$-ary constant-weight codes. While a vast amount of knowledge exists for binary constant-weight codes and $q$-ary constant-weight codes, Recently, these classes of codes have attracted some attention due to several important applications requiring nonbinary alphabets, such as in determining the zero error decision feedback capacity of discrete memoryless channels, multiple access communications, spherical codes for modulation, DNA codes, powerline communications, frequency hopping, and coding for bandwidth-limited channels.

As in the case of binary constant-weight codes, the determination of the maximum size of a constant-composition code or a $q$-ary constant-weight code of length $n$, given constraints on its distance, weight and/or composition, constitutes a central problem in their investigation.

The ring $\mathbb{Z}/q\mathbb{Z}$ is denoted by $\mathbb{Z}_q$. For integers $m \leq n$, the set of integers $\{m, m+1, \ldots, n\}$ is denoted $[m, n]$. The set $[1, n]$ is further abbreviated by $[n]$. A partition $\mathcal{X} = [\lambda_1, \ldots, \lambda_N]$ of integers such that $\lambda_1 \geq \cdots \geq \lambda_N \geq 1$. The $\lambda_i$’s are the parts of the partition. Disjoint set union is denoted by $\sqcup$.

The research of Y. M. Chee and S. Ling is supported in part by the National Research Foundation of Singapore under Research Grant NRF-CRP2-2007-03. The research of Y. M. Chee is also supported in part by the Nanyang Technological University under Research Grant MS8110040.

A. C. H. Ling is with the Department of Computer Science, University of Vermont, Burlington, Vermont, USA 05405 (email: aling@emba.uvm.edu).

The following operations do not affect distance and composition properties of an $(n, d, \overline{w})$-code:
1) reordering the components of $\overline{w}$, and
2) deleting zero components of $\overline{w}$.

Consequently, throughout this paper, attention is restricted to those compositions $\overline{w} = [w_1, \ldots, w_q - 1]$, where $w_1 \geq \cdots \geq w_q - 1 \geq 1$, that is, $\overline{w}$ is a partition. For succinctness, the sum $\sum_{i=1}^{q} w_i$ of all the parts of a partition $\overline{w} = [w_1, \ldots, w_q - 1]$ is denoted by $\sum \overline{w}$.

The focus of this paper is on determining $A_q(n, d, w)$ and $A_q(n, d, \overline{w})$ for those $d, w$ and $\overline{w}$ for which $A_q(n, d, w) = O(n)$ and $A_q(n, d, \overline{w}) = O(n)$.

The Johnson-type bound of Svanström for ternary constant-composition codes [5] Theorem 1] extends easily to the
following (see also [27] Proposition 1.3):  

\textbf{Proposition 1.1 (Johnson Bound):}  
\[ A_q(n, d, [w_1, w_2, \ldots, w_{q-1}]) \leq \left\lfloor \frac{n}{w_1} A_q(n-1, d, [w_1-1, w_2, \ldots, w_{q-1}]) \right\rfloor. \]

The following Johnson-type bound for q-ary constant-weight codes was established in [6] Theorem 10.  

\textbf{Proposition 1.2 (Johnson Bound):}  
\[ A_q(n, d, w) \leq \left\lfloor \frac{n(q-1)}{w} A_q(n-1, d, w-1) \right\rfloor. \]

\textbf{Definition 1.1 (Refinement):} A partition \( \overline{w} = [w_1, \ldots, w_q] \) is a refinement of \( \overline{v} = [v_1, \ldots, v_{q'}] \) (written \( \overline{w} \succ \overline{v} \)) if there exist pairwise disjoint sets \( I_1, \ldots, I_{q'} \subseteq [q] \) satisfying \( \bigcup_{j \in [q']} I_j = [q] \) such that \( \sum_{i \in I_j} w_i = v_j \) for each \( j \in [q'] \).  

Chu et al. [27] made the following observation.  

\textbf{Lemma 1.1:} If \( \overline{w} \succ \overline{v} \), then \( A_q(n, d, \overline{w}) \geq A_q(n, d, \overline{v}). \)

Given \( q \) and \( w \), the condition for \( A_q(n, d, \overline{w}) = O(n) \) to hold can be characterized as follows.  

\textbf{Proposition 1.3:} \( A_q(n, d, \overline{w}) = O(n) \) if and only if \( d \geq 2 \sum \overline{w} - 1 \).  

\textbf{Proof:} \( A_q(n, d, \overline{w}) = O(n) \) when \( d \geq 2 \sum \overline{w} - 1 \) follows easily from the Johnson bound.  

Rödl’s proof [29] of the Erdős-Hanani conjecture [30] implies that \( A_q(n, d, w) = (1 - o(1)) \left( \frac{w - d + 1}{w - d + 1} \right)^{\frac{d}{2}} \), so that \( A_q(n, d, w) = \Omega(n^2) \) for all \( d \leq 2w - 2 \). Therefore, by Lemma 1.1, \( A_q(n, d, \overline{w}) \geq A_q(n, d, \sum \overline{w}) = \Omega(n^2) \) for all \( d \leq 2 \sum \overline{w} - 2 \). \hfill \Box

A similar proof yields:  

\textbf{Proposition 1.4:} \( A_q(n, d, w) = O(n) \) if and only if \( d \geq 2w - 1 \).

\section{II. Quasicyclic Codes}

A code is quasicyclic if there exists an \( \ell \) such that a cyclic shift of a codeword by \( \ell \) places is another codeword. More formally, let \( X = \mathbb{Z}_n \) and define on \( \mathbb{Z}_n^q \) the cyclic shift operator \( T : (u_x)_{x \in X} \mapsto (u_{x+1})_{x \in X} \). A q-ary code \( C \subseteq \mathbb{Z}_n^q \) of length \( n \) is quasicyclic (or more precisely, \( \ell \)-quasicyclic) if it is invariant under \( T^\ell \) for some integer \( \ell \in [n] \). If \( \ell = 1 \), such a code is just a cyclic code.

The following two conditions are necessary and sufficient for a code \( C \) of constant weight \( w \) to have distance \( 2w - 1 \).  

\begin{itemize}
  \item \textbf{(C1)} For any distinct \( u, v \in C \), \( |\text{supp}(u) \cap \text{supp}(v)| \leq 1 \).
  \item \textbf{(C2)} For any distinct \( u, v \in C \), if \( x \in \text{supp}(u) \cap \text{supp}(v) \), then \( u_x \neq v_x \).
\end{itemize}

\textbf{A. Quasicyclic Constant-Composition Codes}

The strategy for proving Main Theorem 1 is to construct optimal \((n, 2 \sum \overline{w} - 1, \overline{w})_q\)-codes (meeting the Johnson bound) that are \( w_1 \)-quasicyclic when \( n \equiv 0 \) (mod \( w_1 \)). Optimal \((n, 2 \sum \overline{w} - 1, \overline{w})_q\)-codes, \( n \not\equiv 0 \) (mod \( w_1 \)), can be obtained easily from those with \( n \equiv 0 \) (mod \( w_1 \)) by lengthening, as in the lemma below.

\textbf{Lemma 2.1 (Lengthening):} If \( A_q(n, 2 \sum \overline{w} - 1, \overline{w}) = [n/w_1] \) and \( n \equiv 0 \) (mod \( w_1 \)), then \( A_q(n+i, 2 \sum \overline{w} - 1, \overline{w}) = [n/w_1] \) for all \( i, 0 \leq i < w_1 \).  

\textbf{Proof:} Let \( C \subseteq \mathbb{Z}_n^X \) be an \((n, 2 \sum \overline{w} - 1, \overline{w})_q\)-code of size \([n/w_1]\). Let \( X' = X \cup \{\infty_1, \ldots, \infty_i\} \), where \( \infty_1, \ldots, \infty_i \not\in X \), and define \( C' \subseteq \mathbb{Z}_n^{X'} \) such that \( C' = \{((c(u))_x)_{x \in X'} : u \in C \} \), where

\[ c(u)_x = \begin{cases} u_x, & \text{if } x \in X \\ 0, & \text{if } x \in \{\infty_1, \ldots, \infty_i\}. \end{cases} \]
Then $C'$ is an $(n+i,2\sum \overline{w}-1,\overline{w})\rho$-code of size $[n/w_1]$. Since $\{(n+i)/w_1\} = [n/w_1]$, $C'$ is optimal by the Johnson bound.

As opposed to lengthening a code, we can also shorten a code by selecting a position $i$, remove those codewords with a nonzero coordinate $i$, and deleting the $i$th coordinate from every remaining codeword.

Let $n \equiv 0 \pmod {w_1}$. A $w_1$-quasicyclic $(n, 2\sum \overline{w}-1, \overline{w})\rho$-code $C'$ of size $n/w_1$ can be obtained by developing a particular vector $g \in \mathbb{Z}_\rho^X$:

$$C' = \{T^{uw_i}(g) : i \in [0, n/w_1-1]\}.$$  

Such a vector $g$ is called a base codeword of the quasicyclic code $C$. The remainder of this section develops criteria for a vector $g \in \mathbb{Z}_\rho^X$ of composition $\overline{w}$ to be a base codeword of a $w_1$-quasicyclic $(n, 2\sum \overline{w}-1, \overline{w})\rho$-code $C$, $n \equiv 0 \pmod {w_1}$.

The conditions (C1) and (C2) may be stated in terms of the base codeword $g$ as follows.

(C3) For $w, x, y, z \in \text{supp}(g)$ such that $w \neq x, y \neq z$, and $\{w, x\} \neq \{y, z\}$, we have:
- if $x - w \equiv 0 \pmod {w_1}$, then $2(x - w) \equiv 0 \pmod {n}$;
- if $y - w \equiv 0 \pmod {w_1}$, then $x - w \neq z - y \pmod {n}$.

(C4) If $g_x = g_y \neq 0$, then $x - y \neq 0 \pmod {w_1}$.

B. Quasicyclic Constant-Weight Codes

Lemma 2.2: Let $n \geq w > 0$ and $q \geq 2$. Then $w|(q-1)n$ if and only if there exist positive integers $\alpha, \beta, \ell$ and $m$ such that $n = \alpha \ell, w = \beta \ell$, and $q - 1 = m \beta$.

Proof: Assume that $w|(q-1)n$. Let $\ell = \gcd(w, n)$, and let $\alpha = n/\ell, \beta = w/\ell$. Then $\gcd(\alpha, \beta) = 1$. Since $w|(q-1)n$, we have $\beta \ell|q-1).$ Hence, $\beta | \ell$. Now let $m = (q - 1)/\beta$.

The converse is obvious.

Suppose that $w|(q-1)n$. By Lemma 2.2, there exist positive integers $\alpha, \beta, \ell$ and $m$ such that $n = \alpha \ell, w = \beta \ell$, and $q - 1 = m \beta$. Our strategy is to construct $\ell$-quasicyclic optimal $(n, 2w-1, w)\rho$-codes of size $(q - 1)n/w = mn/\ell$ (meeting the Johnson bound). In other words, we want to find $m$ vectors, $g^{(1)}, \ldots, g^{(m)} \in \mathbb{Z}_\rho^X$, of weight $w$, such that

$$C = \{T^{e_i}(g^{(i)}) : i \in [0, n/w_1-1] \text{ and } j \in [m]\}$$

is an $(n, 2w-1, w)\rho$-code of size $mn/\ell$. The vectors $g^{(1)}, \ldots, g^{(m)}$ are referred to as base codewords of $C$.

The conditions (C1) and (C2) can be stated in terms of the base codewords $g^{(1)}, \ldots, g^{(m)}$ as follows.

(C5) Let $w, x \in \text{supp}(g^{(i)})$ and $y, z \in \text{supp}(g^{(j)})$ such that $w \neq x, y \neq z$, and $\{w, x\} \neq \{y, z\}$ if $i = j$. Then we have:
- if $x - w \equiv 0 \pmod {\ell}$, then $2(x - w) \equiv 0 \pmod {n}$;
- if $y - w \equiv 0 \pmod {\ell}$, then $x - w \neq z - y \pmod {n}$.

(C6) If $g^{(j)}_x = g^{(j)}_y \neq 0$ and $z \neq y$, then $z - y \neq 0 \pmod {\ell}$, for all $j \in [m]$.

(C7) If $g^{(j)}_x = g^{(j)}_y \neq 0$ (z and $y$ are not necessarily distinct), then $z - y \neq 0 \pmod {\ell}$, for all $i, j \in [m], i \neq j$.

III. A NEW COMBINATORIAL ARRAY

Conditions (C3)–(C4) (respectively, (C5)–(C7)) suggest organizing the elements of $\text{supp}(g)$ (respectively, $\text{supp}(g^{(j)})$) of those quasicyclic constant-composition codes (respectively, constant-weight codes) into a two-dimensional array, with respect to their congruence class modulo $w_1$ (respectively, $\ell$) and the value of their corresponding components in $g$ (respectively, $g^{(1)}, \ldots, g^{(m)}$).

Definition 3.1: Let $\mathbf{X} = [\lambda_1, \ldots, \lambda_N]$ be a partition. A $\mathbf{X}$-array is a $\lambda_1 \times N$ array $B$ with rows indexed by $i \in [\lambda_1]$ and columns indexed by $j \in [N]$, such that

(P1) each cell is either empty or contains a nonnegative integer congruent to its row index modulo $\lambda_1$;

(P2) the number of nonzero cells in column $j$ is $\lambda_j$;

(P3) if $B_i = \{b_{i,j_1}, \ldots, b_{i,N_i}\}$ is the set of entries in row $i$ of $B$, then the differences $b_{i,j} - b_{i,j'}$, $i \in [N]$, $1 \leq j' \neq j \leq N_i$, are all nonzero and distinct.

The scope of $B$ is

$$\sigma(B) = \max_{1 \leq i \leq \lambda_1} \{r_i' : 1 \leq j' \neq j \leq N_i\} \cup \{r_{i,j'} : j \in [N_i]\}.$$  

In particular, if $\lambda_1 = \cdots = \lambda_N = \lambda$, then a $\lambda$-array $B$ has all cells nonzero, and is referred to as a $(\lambda, N)$-array. From the definition, it is easy to see that the entries of a $\mathbf{X}$-array are all distinct.

Example 3.1: A $[3, 2, 2]$-array of scope 15:

$$\begin{array}{ccc}
1 & 7 & 16 \\
2 & 14 \\
3 & \\
\end{array}$$

Example 3.2: A $(2,4)$-array of scope 42:

$$\begin{array}{cccc}
19 & 23 & 35 & 61 \\
0 & 6 & 20 & 30 \\
\end{array}$$

Proposition 3.1: Let $\mathbf{w} = [w_1, \ldots, w_{q-1}]$. If there exists a $\mathbf{w}$-array $B$, then there exists a $w_1$-quasicyclic optimal $(n, 2\sum \overline{w}-1, \overline{w})\rho$-code for all $n \equiv 0 \pmod {w_1}, n \geq 2\sigma(B) + 1$.

Proof: Let $B$ be a $\mathbf{w}$-array and let $C_j$ denote the set of entries in column $j$ of $B, j \in [q-1]$. Define a vector $g \in \mathbb{Z}_\rho^X, n \geq 2\sigma(B) + 1$, as follows:

$$g_x = \begin{cases} 
 j, & \text{if } x \in C_j \\
 0, & \text{otherwise}
\end{cases}$$

Then $g$ has composition $\mathbf{w}$ and satisfies conditions (C3) and (C4). Therefore $g$ is a base codeword of a $w_1$-quasicyclic optimal $(n, 2\sum \overline{w}-1, \overline{w})\rho$-code.
Example 3.3: The \([3, 2, 2]\)-array in Example 3.1 gives the base codeword
\[
g = 111200020000030030^{-17}
\]
for a 3-quasicyclic optimal \((n, 13, [3, 2, 2])_4\)-code when \(n \equiv 0 \pmod{3}\), \(n \geq 33\).

Proposition 3.2: Suppose that \(w = \beta \ell\) and \(q - 1 = m \beta\).
If there exists an \((\ell, q - 1)\)-array \(B\) then there exists an \(\ell\)-quasicyclic optimal \((n, 2w - 1, w)\)-code of size \((q - 1)n/w = mn/\ell\), provided that \(\ell | n\) and \(n \geq 2 \sigma(B) + 1\).

Proof: Let \(B\) be an \((\ell, q - 1)\)-array and let \(C_i\) denote the set of entries in column \(i\) of \(B\), \(i \in [q - 1]\). We define the \(m\) vectors \(g^{(1)}, \ldots, g^{(m)}\) as follows: for \(j \in [m]\) and \(0 \leq z \leq n - 1\),
\[
g^{(j)}_z = \begin{cases} r, & \text{if } z \in C_r \text{ for some } r \in [(j - 1) \beta + 1, j \beta] \\ 0, & \text{otherwise} \end{cases}
\]
(1)

Since the entries of \(B\) are distinct, \(g^{(j)}\) is well-defined. Moreover, the set of nonzero entries of \(g^{(j)}\) is precisely \([(j - 1) \beta + 1, j \beta]\), and by property (P2), each symbol in \([(j - 1) \beta + 1, j \beta]\) occurs exactly \(\ell\) times in \(g^{(j)}\). Therefore, \(g^{(j)} \in \mathbb{Z}_n^{\ell} \) and has weight \(w = \beta \ell\).

We claim that the \(m\) vectors \(g^{(1)}, \ldots, g^{(m)}\) satisfy conditions (C5)–(C7), and hence form the base codewords for an \(\ell\)-quasicyclic optimal \((n, 2w - 1, w)\)-code. The following establishes this claim.

First, suppose that \(i \neq j\). If \(g^{(i)}_z \neq 0\) and \(g^{(j)}_z \neq 0\) are nonzero, then \(g^{(i)}_z \in [(i - 1) \beta + 1, i \beta]\) and \(g^{(j)}_z \in [(j - 1) \beta + 1, j \beta]\). Since \(i \neq j\), we have \(g^{(i)}_z \neq g^{(j)}_z\). Therefore (C7) is satisfied.

Next, suppose that \(z \neq y\) and \(g^{(i)} = g^{(j)} = r \neq 0\). By (1), \(z, y \in C_r\). Since \(z \neq y\), \(z\) and \(y\) must belong to different rows of \(B\). Therefore, \(z \neq y \pmod{\ell}\) by (P1). Thus, \(g^{(1)}, \ldots, g^{(m)}\) satisfy (C6).

Now suppose that \(w, x \in \text{supp}(g^{(i)})\), \(w \neq x\). By (1), there exist \(r_w\) and \(r_x\) such that \(w \in C_{r_w}\) and \(x \in C_{r_x}\). If \(x - w \equiv 0 \pmod{\ell}\), then by (P1), \(x\) and \(w\) are in the same row of \(B\). Therefore,
\[
0 < |x - w| \leq \sigma(B),
\]
and hence,
\[
0 < 2|x - w| \leq 2 \sigma(B) < 1 + 2 \sigma(B) \leq n.
\]
It follows that \(2|x - w| \neq 0 \pmod{n}\).

Let \(w, x \in \text{supp}(g^{(i)})\) and \(y, z \in \text{supp}(g^{(j)})\), where \(w \neq x\), \(y \neq z\) such that \(y - w \equiv 0 \pmod{\ell}\), and if \(i = j\) then \(\{w, x\} \neq \{y, z\}\). We want to show that \(x - w \neq z - y \pmod{\ell}\).

or equivalently,
\[
y - w \neq z - x \pmod{n}.
\]
Again, by (1), \(w, x, y, z\) are entries of \(B\). Moreover, \(w\) and \(y\) are in the same row. We consider two cases.

Case \(w \neq y\)
Since \(0 < |y - w| \leq \sigma(B) < n\), we have \(y - w \neq 0 \pmod{n}\). Therefore, if \(x = z\), then (2) holds. If \(x \neq z\) and both \(x\) and \(z\) are in the same row, then (2) holds by property (P3) of \(B\) and the assumption that \(y \neq z\) and \(n \geq 2 \sigma(B) + 1\). If \(x\) and \(z\) are in different rows, then by (P1), \(z - x \neq 0 \pmod{\ell}\). Since \(y - w \equiv 0 \pmod{\ell}\) and \(\ell | n\), (2) follows.

Case \(w = y\)
We claim that \(i = j\). Indeed, assume that \(y \in C_{r_y}\) and \(w \in C_{r_w}\). Then \(r_y \in [(j - 1) \beta + 1, j \beta]\) and \(r_w \in [(i - 1) \beta + 1, i \beta]\). Hence, if \(i \neq j\), then \(r_y \neq r_w\). Therefore, there are two entries in different columns of \(B\) that have the same value \(y\), which is a contradiction. Hence, \(i = j\). Since \(\{w, x\} \neq \{y, z\}\), we have \(x \neq z\). Therefore, (2) holds.

Consequently, \(g^{(1)}, \ldots, g^{(m)}\) satisfy (C5).

Example 3.4: The \((2, 4)\)-array of scope 42 in Example 3.2 gives \(g^{(1)}\) and \(g^{(2)}\), where
\[
g^{(1)} = \begin{cases} 1, & \text{if } z \in \{0, 19\} \\ 2, & \text{if } z \in \{6, 23\} \\ 0, & \text{otherwise} \end{cases}
\]
\[
g^{(2)} = \begin{cases} 3, & \text{if } z \in \{20, 35\} \\ 4, & \text{if } z \in \{30, 61\} \\ 0, & \text{otherwise} \end{cases}
\]
In this case, \(q = 5\), \(w = 4\), \(\beta = 2\), \(\ell = 2\), and \(m = 2\). The vectors \(g^{(1)}\) and \(g^{(2)}\) form the base codewords of a 2-quasicyclic optimal \((n, 7, 4)\)-code when \(n\) is even and \(n \geq 85 = 2 \times 42 + 1\).

In view of Proposition 3.1 and Proposition 3.2, to prove Main Theorem 1 and Main Theorem 2, it suffices to construct a \(\overline{X}\)-array for every partition \(\overline{X}\).

IV. Generalized Difference Triangle Sets

In this section, the concept of difference triangle sets is generalized and used to produce \(\overline{X}\)-arrays. We begin with the definition of a difference triangle set.

Definition 4.1: An \((I, J)\)-difference triangle set (D\(\overline{X}\)S) is a set \(A = \{A_1, \ldots, A_J\}\), where \(A_i = \{a_{i,1}, \ldots, a_{i,J}\}, 0 = a_{i,1} < \cdots < a_{i,J}\), are lists of integers such that the differences \(a_{i,j} - a_{i,j'}, i \in \{I\}, j \neq j' \neq J\) are all distinct.

Example 4.1: A \((3, 4)\)-D\(\overline{X}\)S:

\[
\{(0, 1, 10, 18), \{0, 2, 7, 13\}, \{0, 3, 15, 19\}\}
\]
The corresponding differences are displayed in triangular arrays below:

\[
\begin{array}{cccc}
1 & 10 & 18 & \\
9 & 17 & 5 & 11 \\
8 & 6 & 4 & \\
\end{array}
\]
The scope of an \((I, J)\)-D\(\overline{X}\)S \(A = \{A_1, \ldots, A_J\}\) is
\[
m(A) = \max\{a \in A\}.
\]
Difference triangle sets with scope as small as possible are often required for applications. Define

\[ M(I,J) = \min \{ \lambda : A \text{ is an } (I,J)-\Delta \lambda \} \]

Difference triangle sets were introduced by Kløve [32, 33] and have numerous applications [34–40]. A \((1,J)-\Delta \lambda\) is known as a Golomb ruler with \(J\) marks.

We generalize difference triangle sets as follows.

**Definition 4.2:** Let \(\mathcal{J} = \{J_1, \ldots, J_I\}\) be a partition. A set \(A = \{A_1, \ldots, A_I\}\) with \(A_i = \{a_{i,1}, \ldots, a_{i,J_i}\}, 0 = a_{i,1} < \cdots < a_{i,J_i}\), is a \(\mathcal{J}\)-generalized difference triangle set (GD\(\Delta\)S) if the differences \(a_{i,j} - a_{i,j'}\), \(i \in [I]\), \(1 \leq j' \neq j \leq J_i\), are all distinct.

Thus, a GD\(\Delta\)S is similar to a D\(\Delta\)S, but allowing the sets to be of different sizes. In particular, if \(J_1 = \cdots = J_I = J\), then a \(\mathcal{J}\)-GD\(\Delta\)S is an \((I,J)-\Delta\lambda\). The scope of a GD\(\Delta\)S \(A = \{A_1, \ldots, A_I\}\) is defined similarly as for a D\(\Delta\)S:

\[ m(A) = \max_{A \in A} \{a : A \in A\} \]

We now relate \(\mathcal{J}\)-GD\(\Delta\)S to \(\lambda\)-arrays. Let \(\lambda = [\lambda_1, \ldots, \lambda_N]\) be a partition. The Ferrers diagram of \(\lambda\) is an array of cells with \(N\) left-justified rows and \(\lambda_i\) cells in row \(i\). The conjugate of \(\lambda\) is the partition \(\lambda^r = [\lambda_1^r, \ldots, \lambda_N^r]\), where \(\lambda_i^r\) is the number of parts of \(\lambda\) that are at least \(j\). \(\lambda^r\) can also be obtained by reflecting the Ferrers diagram of \(\lambda\) along its main diagonal. Conjugation of partitions is an involution.

**Example 4.2:** The Ferrers diagrams of the partition \([5,3,3,2]\) and its conjugate \([4,4,3,1,1]\) are shown respectively below:

\[
\begin{array}{cccc}
5 & 3 & 3 & 2 \\
3 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
\end{array}
\]

**Proposition 4.1:** Let \(\lambda = [\lambda_1, \ldots, \lambda_N]\) be a partition. If there exists a \(\lambda^r\)-GD\(\Delta\)S of scope \(s\), then there exists a \(\lambda\)-array of scope at most \(s\lambda_1\).

**Proof:** Let \(\lambda^r = [\lambda_1^r, \ldots, \lambda_N^r]\) and let \(A = \{A_1, \ldots, A_N\}\) be a \(\lambda^r\)-GD\(\Delta\)S of scope \(s\). Construct a \(\lambda \times N\) array \(B\) as follows: If \(A_i = \{a_{i,1}, \ldots, a_{i,N}\}\), then the \((i,j)\)th cell of \(B\), \(i \in [\lambda_i]\), \(j \in [N]\), contains \(b_{i,j} = a_{i,j} \lambda_1 + (i \mod \lambda_i)\) if \(j \in [\lambda_i^r]\), and empty otherwise. Then the filled cells of \(B\) take the shape of the Ferrers diagram of \(\lambda^r\). Thus, the number of nonempty cells in column \(j\) of \(B\) is precisely \(\lambda_j\). It is also easy to see that each entry in row \(i\) of \(B\) is congruent to \(i \mod \lambda_i\). The differences \(b_{i,j} - b_{i,j'}\) are all distinct because the differences \(a_{i,j} - a_{i,j'}\) are all distinct in the GD\(\Delta\)S \(A\). Moreover, all of these differences are at most \(s\lambda_1\). Finally, for any \(i \in [\lambda_1]\) and \(j \in [\lambda_i^r]\),

\[
\left\lfloor \frac{b_{i,j}}{2} \right\rfloor \leq \frac{s\lambda_1 + (\lambda_1 - 1)}{2} \leq \frac{s\lambda_1 + \lambda_1}{2} \leq s\lambda_1.
\]

Therefore \(B\) is a \(\lambda\)-array of scope at most \(s\lambda_1\).

**Corollary 4.1:** If there exists a \((\lambda, N)\)-GD\(\Delta\)S of scope \(s\), then there exists a \((\lambda, N)\)-array of scope at most \(s\lambda\).

**Example 4.3:** Since \([3,3,2,2]^* = [4,4,2]\), we can construct a \([3,3,2,2]\)-array from a \([4,4,2]\)-GD\(\Delta\)S via the proof of Proposition 4.1. If the \([4,4,2]\)-GD\(\Delta\)S is \(A = \{0,1,10,18\}, \{0,2,7,13\}, \{0,3\}\), the \([3,3,2,2]\)-array obtained is

\[
\begin{array}{cccc}
1 & 4 & 31 & 55 \\
2 & 8 & 23 & 41 \\
0 & 9 & 45 & 57 \\
\end{array}
\]

This array has scope 54.

**Example 4.4:** From the \((3,4)\)-GD\(\Delta\)S \(A = \{\{0,1,10,18\}, \{0,2,7,13\}, \{0,3,15,19\}\}\), we can construct the following \((3,4)\)-array via the proof of Proposition 4.1

\[
\begin{array}{cccc}
1 & 4 & 31 & 55 \\
2 & 8 & 23 & 41 \\
0 & 9 & 45 & 57 \\
\end{array}
\]

This array has scope 57.

**V. PROOFS OF THE MAIN THEOREMS**

In this section, we use Golomb rulers to construct GD\(\Delta\)S and provide proofs to Main Theorem 1 and Main Theorem 2.

Let \(\varphi(x)\) denote the smallest prime power not smaller than \(x\). Atkinson et al. [40] Lemma 2 proved the following.

**Theorem 5.1:** \(M(1,J) \leq (J-1)\varphi(J-1)\).

**Proposition 5.1:** For any partition \(\mathcal{J} = [J_1, \ldots, J_I]\), there exists a \(\mathcal{J}\)-GD\(\Delta\)S of scope at most \((\sum \mathcal{J} - 1)\varphi(\sum \mathcal{J} - 1)\).

**Proof:** By Theorem 5.1, there exists a Golomb ruler \(\{R_I\}\) of \(\sum \mathcal{J}\) marks and scope \(m(\{R_I\}) \leq (\sum \mathcal{J} - 1)\varphi(\sum \mathcal{J} - 1)\). Partition \(R\) into \(I\) subsets, \(R = R_1 \cup \cdots \cup R_I\), where \(|R_i| = J_i, i \in [I]\). Suppose \(R_i = \{r_{i,1}, \ldots, r_{i,J_i}\}\), where \(0 \leq r_{i,1} < \cdots < r_{i,J_i}\). For each \(i \in [I]\), let \(A_i = \{a_{i,1}, \ldots, a_{i,J_i}\}\), where \(a_{i,j} = r_{i,j} - r_{i,1}, j \in [J_i]\). Then the set \(A = \{A_1, \ldots, A_I\}\) forms a \(\mathcal{J}\)-GD\(\Delta\)S of scope

\[ m(A) \leq m(\{R\}) \leq (\sum \mathcal{J} - 1)\varphi(\sum \mathcal{J} - 1). \]

The following corollary is immediate.

**Corollary 5.1:** For any \(I > 0\) and \(J > 0\), there exists an \((I,J)\)-GD\(\Delta\)S of scope at most \((IJ - 1)\varphi(IJ - 1)\).
A. Proof of Main Theorem 1

Let \( \overline{w} = [w_1, \ldots, w_{q-1}] \) be a partition and consider \( \overline{w}^* = [w_1^*, \ldots, w_{q}^*] \). By Proposition 5.1 there exists a \( \overline{w}^* \)-GDS of scope at most \( (\sum \overline{w} - 1)\varphi(\sum \overline{w} - 1) \). Therefore, by Proposition 4.1 there exists a \( \overline{w} \)-array of scope at most \( w_1(\sum \overline{w} - 1)\varphi(\sum \overline{w} - 1) \). Finally, Proposition 5.1 guarantees the existence of a \( w_1 \)-quasicyclic optimal \((n, 2\sum \overline{w} - 1, \overline{w}_3)\sigma\) code of size \( n/w_1 \) for all \( n \equiv 0 \pmod{w_1}, \) \( n \geq 2w_1(\sum \overline{w} - 1)\varphi(\sum \overline{w} - 1) + 1 \). This, together with Lemma 2.1 proves Main Theorem 1.

B. Proof of Main Theorem 2

Suppose \( w|(q-1)n \). Then by Lemma 2.2 let \( w = \beta\ell \), where \( \beta|(q-1) \). By Corollary 5.1 there exists an \((\ell, q-1)\)-D\( A \) of scope at most \((\ell(q-1) - 1)\varphi(\ell(q-1) - 1) \). Therefore, by Corollary 4.1 there exists an \((\ell, q-1)\)-array of scope at most \((\ell(q-1) - 1)\varphi(\ell(q-1) - 1) \). Finally, Proposition 5.2 guarantees the existence of an \((\ell, q-1)\)-quasicyclic optimal \((n, 2\ell - 1, \ell_q)\) code of size \((q-1)n/w \) for all \( n \equiv 0 \pmod{\ell}, \) \( n \geq 2\ell(q-1)\varphi(q-1) + 1 \). This proves Main Theorem 2.

In particular, by taking \( \beta = 1 \) and \( \beta = w \), respectively, we have the following results:

(i) There exists a \( w \)-quasicyclic optimal \((n, 2\ell - 1, \ell)\) code for all \( n \equiv 0 \pmod{w} \), \( n \geq 2\ell w(q-1) - 1 \).

(ii) If \( w|(q-1) \), then there exists a cyclic optimal \((n, 2\ell - 1, \ell)\) code for all \( n \geq 2(q-1)w(q-2) + 1 \).

VI. Resolution of an Open Problem of Etzion

A set system is a pair \( S = (X, B) \), where \( X \) is a finite set of points, and \( B \subseteq 2^X \). The elements of \( B \) are called blocks. The order of \( S \) is the number of points, \( |X| \). If \( |B| = k \) for all \( B \in B, \) then \( S \) is said to be \( k \)-uniform. Let \( A \subseteq 2^X \). A transverse of \( A \) is set \( T \subseteq X \) such that \(|T \cap A| = 1 \) for all \( A \in A \). Hanani \[41\] introduced the following generalization of \( t \)-designs.

**Definition 6.1:** An \( H(n, q, w, t) \) design is a triple \((X, G, B)\), where \((X, B)\) is a \( w \)-uniform set system of order \( nq \), \( G = \{G_1, \ldots, G_n\} \) is a partition of \( X \) into \( n \) sets, each of cardinality \( q \), such that

(i) \( B \) is a transverse of \( G \) for all \( B \in B \); and
(ii) each \( t \)-element transverse of \( G \) is contained in precisely one block of \( B \).

From an \( H(n, q, w, t) \) design, we can form a constant-weight code \( C \subseteq Z_q^{n+1} \) as follows. Let \( G_i = \{\gamma_{1,i}, \gamma_{2,i}, \ldots, \gamma_{q,i}\} \), where \( 0 \notin G_i \). The code \( C \) has a codeword for each block. Assume \( B = \{b_1, b_2, \ldots, b_w\} \) is a block of \( B \) (this block is denoted by \( \{i_1, j_1\}, \{i_2, j_2\}, \ldots, \{i_w, j_w\} \)). We form the codeword \( u \in C \) corresponding to \( B \) as follows: for \( i \in [n] \),

\[
  u_i = \begin{cases} 
    j, & \text{if } b_r = \gamma_{j,i} \text{ for some } r \in [w] \\
    0, & \text{otherwise.}
  \end{cases}
\]

The distance of \( C \) is at least \( w-t+1 \). If \( C \) has distance \( 2(w-t) + 1 \), Etzion \[31\] calls the \( H(n, q, w, t) \) design, from which \( C \) is constructed, a generalized Steiner system \( GS(t, w, n, q) \).

It is not hard to verify that a \( GS(t, w, n, q) \) contains exactly \( q^t \left( \begin{array}{c} n \\ t \end{array} \right) \) blocks. By the Johnson bound, we have

\[
  A_{q+1}(n, 2(w-t) + 1, w) = q^t \left( \begin{array}{c} n \\ t \end{array} \right).
\]

It follows from the above construction that if a \( GS(t, w, n, q) \) exists, then

\[
  A_{q+1}(n, 2(w-t) + 1, w) = q^t \left( \begin{array}{c} n \\ t \end{array} \right).
\]

The next result establishes the converse when \( \left( \begin{array}{c} n \\ t \end{array} \right) \). Then a \( GS(t, w, n, q) \) exists if

\[
  A_{q+1}(n, 2(w-t) + 1, w) = q^t \left( \begin{array}{c} n \\ t \end{array} \right).
\]

**Proof:** Let \( C \) be an (optimal) \((n, 2(w-t) + 1, w)\) code of size \( \left( \begin{array}{c} n \\ t \end{array} \right) \). Define

\[
  X = \{(i, j) : i \in [n] \text{ and } j \in [q]\}
\]

\[
  G = \{G_i : i \in [n]\},
\]

where \( G_i = \{(i, j) : j \in [q]\} \). We associate each codeword \( u \in C \) a block \( B^u \subseteq X \) as follows:

\[
  B^u = \{(i, j) : u_i = j, i \in [n], j \in [q]\}.
\]

Finally, let \( B = \{B^u : u \in C\} \).

We claim that \((X, G, B)\) is a \( GS(t, w, n, q) \). Indeed, \( |B| = w \) for all \( B \in B, \) and \( |B \cap G_i| \leq 1 \) for all \( B \in B \) and \( i \in [n] \). Hence, it remains to show that any \( t \)-element transverse of \( G \) is contained in exactly one block of \( B \). Suppose \( B^u \) and \( B^v \) are two different blocks containing a particular \( t \)-element transverse of \( G \). Then \( |\text{supp}(u) \cap \text{supp}(v)| \geq t \), implying \( d_H(u, v) \leq 2(w-t) < 2(w-t) + 1 \), a contradiction. Therefore, any \( t \)-element transverse of \( G \) is contained in at most one block, and hence in exactly one block, since \( |B| = |C| = q^t \left( \begin{array}{c} n \\ t \end{array} \right) \). \( \blacksquare \)

**Corollary 6.1:** Suppose that \( q^t \left( \begin{array}{c} n \\ t \end{array} \right) \). Then there exists a \( GS(t, w, n, q) \) if and only if

\[
  A_{q+1}(n, 2(w-t) + 1, w) = q^t \left( \begin{array}{c} n \\ t \end{array} \right).
\]

Etzion \[31\] Problem 7] raised the following as an open problem for further research.

**Problem 6.1 (Etzion):** Given \( k \) and \( w \), show that there exists an \( n_0 \) such that for all \( n \geq n_0 \), where \( w|nk \), a \( GS(1, w, n, k) \) exists.

The following result, which is a direct consequence of Main Theorem 2 and Corollary 6.1, solves Problem 6.1.
Theorem 6.1: There exists a GS(1, w, n, k) for all sufficiently large n satisfying w|nk.

Proof: By Main Theorem 2, we have

\[ A_{k+1}(n, 2w - 1, w) = kn/w, \]

for all sufficiently large n satisfying w|kn. It follows immediately from Corollary 6.1 that there also exists a GS(1, w, n, k) for all sufficiently large n satisfying w|kn.

\[ \text{VII. Explicit Bounds} \]

Main Theorem 1 and Main Theorem 2 are asymptotic statements: the hypothesis that \( n \) is sufficiently large must be satisfied. But how large must \( n \) be? More precisely, for a partition \( \mathbf{w} = [w_1, \ldots, w_{q-1}] \) and a positive integer \( w \), define

\[ N_{ccc}(\mathbf{w}) = \min \left\{ n_0 : A_q(n, 2\sum \mathbf{w}-1, \mathbf{w}) = \left\lfloor \frac{n}{w_1} \right\rfloor \text{ for all } n \geq n_0 \right\}, \]

and

\[ N_{cwc}(w) = \min \left\{ n_0 : A_q(n, 2w - 1, w) = \frac{(q-1)n}{w} \text{ for all } n \geq n_0 \right\}, \]

satisfying \( w|(q-1)n \).

We give explicit bounds on \( N_{ccc}(\mathbf{w}) \) and \( N_{cwc}(w) \) in this section.

A. Bounds on \( N_{ccc}(\mathbf{w}) \)

The proof of Main Theorem 1 in Section VII.A shows that

\[ N_{ccc}(\mathbf{w}) \leq 2w_1(\sum \mathbf{w}-1)\varphi(\sum \mathbf{w}-1) + 1. \]  \hfill (3)

By Bertrand’s postulate, \( \varphi(x) \leq 2x \) for all \( x \geq 1 \). For \( x \) sufficiently large, better asymptotic bounds on \( \varphi(x) \) exist (see for example, \[52\]), but we are after quantifiable bounds. This implies

\[ N_{ccc}(\mathbf{w}) \leq 4w_1(\sum \mathbf{w}-1)^2 + 1. \]

We now prove a lower bound on \( N_{ccc}(\mathbf{w}) \).

Proposition 7.1: Let \( \mathbf{w} = [w_1, \ldots, w_{q-1}] \) be a partition. If \( w_1n \) and there exists an \( (n, 2\sum \mathbf{w}-1, \mathbf{w})_q \)-code of size \( n/w_1 \), then \( n \geq w_1^2k(k-1) + w_1 \), where \( k = |\sum \mathbf{w}/w_1| \). In particular, when \( w_1 = w_2 = \cdots = w_{q-1} \), we have \( n \geq w_1^2 + w_1^2(q-1)(q-2) \).

Proof: Let \( C = \{u^{(1)}, \ldots, u^{(n/w_1)}\} \) be an \( (n, 2\sum \mathbf{w}-1, \mathbf{w})_q \)-code of size \( n/w_1 \). Then \( C \) can be regarded as an \( n/w_1 \times n \) matrix \( C \), whose \( i \)th row is \( u^{(i)}, i \in [n/w_1] \). Let \( N_i \) be the number of nonzero entries in column \( i \) of \( C \). Then \( \sum_{i=1}^n N_i = (n \sum \mathbf{w})/w_1 \). In each column of \( C \), we associate each pair of distinct nonzero entries with the pair of rows that contain these entries. There are \( \binom{N_i}{2} \) such pairs of nonzero entries in column \( i \) of \( C \). Therefore, there are \( \sum_{i=1}^n \binom{N_i}{2} \) such pairs in all the columns of \( C \). Since there are no pairs of distinct codewords in \( C \) whose supports intersect in two elements, the \( \sum_{i=1}^n \binom{N_i}{2} \) pairs of rows associated with the \( \sum_{i=1}^n \binom{N_i}{2} \) pairs of distinct nonzero entries are also all distinct. Hence,

\[ \sum_{i=1}^n \binom{N_i}{2} \leq \frac{|C|}{2} = \left( \frac{n}{w_1} \right)^2, \]

or equivalently,

\[ \sum_{i=1}^n N_i(N_i - 1) \leq \frac{n(n - w_1)}{w_1^2}. \] \hfill (4)

Since \( k = |\sum \mathbf{w}/w_1| = \left\lfloor (n \sum \mathbf{w})/w_1 \right\rfloor/n \), there exists \( r \in [0, n-1] \) such that

\[ n \sum \mathbf{w}/w_1 = kn + r. \]

As \( \sum_{i=1}^n N_i = (n \sum \mathbf{w})/w_1 \) we have

\[ \sum_{i=1}^n N_i(N_i - 1) \geq r(k+1)k + (n-r)k(k-1) \geq nk(k-1). \] \hfill (5)

From (4) and (5), we have

\[ n(n - w_1)/w_1^2 \geq nk(k-1), \]

giving \( n \geq w_1^2k(k-1) + w_1 \).

Corollary 7.1:

\[ (\sum \mathbf{w})^2 - w_1(\sum \mathbf{w}-1) \leq N_{ccc}(\mathbf{w}) \leq 4w_1(\sum \mathbf{w}-1)^2 + 1. \]

The upper and lower bounds on \( N_{ccc}(\mathbf{w}) \) in Corollary 7.1 differ approximately by a factor of \( 4w_1 \).

B. Bounds on \( N_{cwc}(w) \)

The proof of Main Theorem 2 in Section VII.B shows that

\[ N_{cwc}(w) \leq 2w(w(q-1) - 1)^2 + 1. \]

For constant-weight codes, the following result of Etzion \[51\] gives \( N_{cwc}(w) \geq (w-1)(q-1) + 1 \).
(C8) \[ \text{supp}(u, v) = 0 \text{ if } u \xleftarrow{T} g(i) \text{ and } v \xleftarrow{T} g(i) \text{ for some } i. \]

(C9) \[ \text{supp}(u, v) \leq 1 \text{ if } u \xleftarrow{T} g(i) \text{ and } v \xleftarrow{T} g(j) \text{ for } i \neq j. \]

We observe that (C8) holds immediately if for every \( i \in [q - 1], g(i) \) is chosen so that \( \text{supp}(g(i)) \) contains \( w \) elements which are congruent to \( 0, 1, \ldots, w - 1 \pmod{w} \), respectively.

**Theorem 7.1:** If \( w|n \) and \( n \geq w((w - 1)(q - 2) + 1) \), then \( A_q(n, 2w - 1, w) = (q - 1)n/w \).

**Proof:** It suffices to show that there exists an \((n, 2w - 1, w)_q\)-code of size \( (q - 1)n/w \) for any \( n \geq w((w - 1)(q - 2) + 1), n \equiv 0 \pmod{w} \). We construct \( q - 1 \) base codewords \( g^{(1)}, \ldots, g^{(w - 1)} \) for such a code as follows. For \( i \in [q - 1], g(i) \in \{0, i\}^n \) satisfies

\[
\text{supp}(g(i)) = \{0, 1 + (i-1)w, 2 + 2(i-1)w, \ldots, \}
\]

\[
(w - 1) + (w-1)(i-1)w). \quad (6)
\]

Condition (C8) is satisfied immediately. It remains to show that these \( q - 1 \) base codewords satisfy (C9). We prove this by contradiction. Assume that there exist \( u = T^{kw} g(i) \) and \( v = T^{lw} g(j) \), \( i \neq j \), so that \( \text{supp}(u, v) \geq 2 \). Suppose that \( a, b \in \text{supp}(u, v) \) and \( a = x \pmod{w} \), \( b = y \pmod{w} \). By (6) we have

\[
a = x + (i-1)w + kw \pmod{n},
\]

\[
x = x + (j-1)w + lw \pmod{n},
\]

and

\[
b = y + y(i-1)w + kw \pmod{n},
\]

\[
y = y + y(j-1)w + lw \pmod{n},
\]

where the terms \( kw \) and \( lw \) result from the cyclic shift operations applied on \( g(i) \) and \( g(j) \). These equations imply

\[
xw(i-j) + (k-l)w \equiv 0 \pmod{n}
\]

and

\[
yw(i-j) + (k-l)w \equiv 0 \pmod{n},
\]

which together yield

\[
(x-y)(i-j) \equiv 0 \pmod{n/w}.
\]

However, since \( 0 \leq x \neq y \leq w - 1 \) and \( 1 \leq i \neq j \leq q - 1 \), we have

\[
0 < |(x-y)(i-j)| \leq (w - 1)(q - 2) < n/w,
\]

as \( n \geq w(1 + (w - 1)(q - 2)) \). Thus, (7) and (8) lead to a contradiction.

### VIII. Tables for Small-Weight Constant-Composition Codes

In this section, we provide two tables of exact values of \( A_q(n, 2 \sum w - 1, \sum w) \) with \( \sum w \leq 6 \), for almost all \( n \). The only undetermined values in this range are \( A_q(n, 11, [1, 1, 1, 1, 1, 1]) \) when \( n \in \{33, 34\} \). The following (trivial) upper bound happens to be very useful when we build up the tables, as it is often tight for codes of small lengths.

**Lemma 8.1:** \( A_q(n, 2 \sum w - 1, \sum w) \leq A_2(n, 2 \sum w - 2, \sum w) \).

Table I provides the base codewords for quasicyclic optimal codes of sufficiently large lengths. For succinctness, we do not indicate trailing zeros at the end of each base codeword. Therefore, the base codeword 1203, say, should be interpreted as \( 12030 \). In order to construct these base codewords, we use either optimal Golomb rulers or a simple computer search to establish the best \( \lambda \)-array corresponding to the codes. Table II includes the sizes of optimal codes with small length \( n \). These two tables together give an almost complete solution for the sizes of optimal constant-composition codes of weight at most six.

In Table III, if a cell is empty, then it means that the corresponding size is already determined in Table I. The upper bound for the sizes of codes comes from either the Johnson bound or Lemma 8.1 whichever is smaller. The lower bounds come from optimal codes constructed by hand or by a hill-climbing algorithm. We refer the interested reader to the Appendix for a complete description of these optimal codes. We note that the values of \( A_q(n, 2(w_1 + w_2) - 1, [w_1, w_2]) \) are included for completeness although it has been determined earlier by Östergård and Svanström [6], Theorem 8.

Table III gives the exact value of \( N_{c\!c\!c}(\mathcal{W}) \) for all \( \mathcal{W} \) such that \( \sum \mathcal{W} \leq 6 \), except when \( \mathcal{W} = [1, 1, 1, 1, 1, 1] \). We compare these values with bounds on \( N_{c\!c\!c}(\mathcal{W}) \) given by [3] and Proposition 7.1. There is a large gap between these bounds. It would be interesting to close this gap.

### IX. Conclusion

The exact sizes of optimal constant-composition and constant-weight codes having linear size are determined for all such codes of sufficiently large lengths. In the course of

<table>
<thead>
<tr>
<th>Weight</th>
<th>Distance</th>
<th>Composition</th>
<th>( N_{c!c!c}(\mathcal{W}) ) from ( \sum w ) and Proposition 7.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>([1, 1])</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>([2, 1])</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>([3, 1])</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>([4, 1])</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>([5, 1])</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
<td>([6, 1])</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>([7, 1])</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>17</td>
<td>([8, 1])</td>
<td>17</td>
</tr>
<tr>
<td>10</td>
<td>19</td>
<td>([9, 1])</td>
<td>19</td>
</tr>
<tr>
<td>11</td>
<td>21</td>
<td>([10, 1])</td>
<td>21</td>
</tr>
<tr>
<td>12</td>
<td>23</td>
<td>([11, 1, 1, 1])</td>
<td>23</td>
</tr>
<tr>
<td>13</td>
<td>25</td>
<td>([12, 1])</td>
<td>25</td>
</tr>
<tr>
<td>14</td>
<td>27</td>
<td>([13, 1, 1, 1])</td>
<td>27</td>
</tr>
<tr>
<td>15</td>
<td>29</td>
<td>([14, 1])</td>
<td>29</td>
</tr>
<tr>
<td>16</td>
<td>31</td>
<td>([15, 1, 1, 1])</td>
<td>31</td>
</tr>
<tr>
<td>17</td>
<td>33</td>
<td>([16, 1])</td>
<td>33</td>
</tr>
<tr>
<td>18</td>
<td>35</td>
<td>([17, 1, 1, 1])</td>
<td>35</td>
</tr>
</tbody>
</table>
### TABLE I

Linear size optimal \((n, 2 \sum w - 1, \overline{w})\)-codes of weight at most six

<table>
<thead>
<tr>
<th>Weight</th>
<th>Distance</th>
<th>Composition</th>
<th>Base codeword</th>
<th>Condition on length (n)</th>
<th>Size</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>[2, 1]</td>
<td>112</td>
<td>(n \geq 5)</td>
<td>([n/2])</td>
<td>Trivial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1, 1, 1]</td>
<td>1203</td>
<td>(n \geq 7)</td>
<td>(n)</td>
<td>[18]</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>[3, 1]</td>
<td>1112</td>
<td>(n \geq 7)</td>
<td>([n/3])</td>
<td>Trivial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2, 2]</td>
<td>112002</td>
<td>(n \geq 10)</td>
<td>([n/2])</td>
<td>This paper</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2, 1, 1]</td>
<td>112003</td>
<td>(n \geq 10)</td>
<td>([n/2])</td>
<td>Refinement of [2, 2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1, 1, 1, 1]</td>
<td>12003040</td>
<td>(n \geq 13)</td>
<td>(n)</td>
<td>This paper</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>[4, 1]</td>
<td>11112</td>
<td>(n \geq 9)</td>
<td>([n/4])</td>
<td>Trivial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[3, 2]</td>
<td>110200020001</td>
<td>(n \geq 15)</td>
<td>([n/3])</td>
<td>This paper</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[3, 1, 1]</td>
<td>110200030001</td>
<td>(n \geq 15)</td>
<td>([n/3])</td>
<td>Refinement of [3, 2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2, 2, 1]</td>
<td>100120000203</td>
<td>(n \geq 18)</td>
<td>([n/2])</td>
<td>This paper</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2, 1, 1, 1]</td>
<td>100120000304</td>
<td>(n \geq 18)</td>
<td>([n/2])</td>
<td>Refinement of [2, 2, 1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1, 1, 1, 1]</td>
<td>1200300000405</td>
<td>(n \geq 23)</td>
<td>(n)</td>
<td>This paper</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>120030000000405</td>
<td>(n = 21)</td>
<td>(n)</td>
<td>This paper</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>[5, 1]</td>
<td>111112</td>
<td>(n \geq 11)</td>
<td>([n/5])</td>
<td>Trivial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[4, 2]</td>
<td>1111200002</td>
<td>(n \geq 20)</td>
<td>([n/4])</td>
<td>This paper</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[3, 1]</td>
<td>1111200003</td>
<td>(n \geq 20)</td>
<td>([n/4])</td>
<td>Refinement of [4, 2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[3, 2]</td>
<td>111200020002</td>
<td>(n \geq 21)</td>
<td>([n/3])</td>
<td>This paper</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[3, 1, 1]</td>
<td>111200030002</td>
<td>(n \geq 21)</td>
<td>([n/3])</td>
<td>Refinement of [3, 3]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2, 2, 1]</td>
<td>111200030004</td>
<td>(n \geq 21)</td>
<td>([n/5])</td>
<td>This paper</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2, 2, 1, 1]</td>
<td>1120020300000003</td>
<td>(n \geq 21)</td>
<td>([n/5])</td>
<td>This paper</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2, 1, 1, 1]</td>
<td>112002030000004</td>
<td>(n \geq 30\text{or }n = 26)</td>
<td>([n/2])</td>
<td>Refinement of [2, 2, 2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1, 1, 1, 1, 1]</td>
<td>1200300004000005</td>
<td>(n \geq 30\text{or }n = 26)</td>
<td>([n/2])</td>
<td>Refinement of [2, 2, 2]</td>
</tr>
</tbody>
</table>

### TABLE II

Sizes of some small optimal constant-composition codes with \(d = 2 \sum w - 1\)

| \(\overline{w}\) | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
|------------------|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| [1, 1]           |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| [2, 1]           |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| [1, 1, 1]        |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| [2, 2]           |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| [2, 1, 1]        |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| [1, 1, 1, 1]     |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| [3, 1]           |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| [3, 2]           |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| [3, 1, 1]        |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| [2, 2, 1]        |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| [2, 1, 1, 1]     |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| [1, 1, 1, 1, 1]  |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | establish these results, we introduced several new concepts, including that of generalized difference triangle sets and showed how they can be constructed from Golomb rulers. The results obtained in this paper solve an open problem of Etzion.

**APPENDIX**

Only codes of size at least five are listed here. Those optimal codes of size four or less can be constructed easily by hand.

### A. Weight Four Codes

1) An Optimal \((10, 7, [1, 1, 1, 1])_5\)-code:

2) An Optimal \((11, 7, [1, 1, 1, 1])_5\)-code:

3) An Optimal \((12, 7, [1, 1, 1, 1])_5\)-code:

### B. Weight Five Codes

1) An Optimal \((15, 9, [2, 2, 1])_4\)-code:

2) An Optimal \((16, 9, [2, 2, 1])_4\)-code:

3) An Optimal \((17, 9, [2, 2, 1])_4\)-code:
00210020000000103 2010100003200000 0003000001202010 000221030010002 01000202001300 120000120000030
007) An Optimal (16, 9, [2, 2, 1])_4-code:
Lengthening of an optimal (15, 9, [2, 1, 1])_4-code.
3) An Optimal (17, 9, [2, 2, 1])_4-code:
003010020000020010 000302100000200020 100000310000200002 002002100000200030 200000012030000001 000100032000002001 021002000002000000
4) An Optimal (n, 9, [2, 1, 1, 1])_5-code, n ∈ [15, 17]:
Refinement of an optimal (n, 9, [2, 1, 1])_4-code, n ∈
[15, 17].
5) An Optimal (n, 9, [[1, 1, 1, 1, 1]]_6-code, n ∈ [15, 18]:
Refinement of an optimal (n, 9, [2, 1, 1, 1])_4-code, n ∈
[15, 18].
6) An Optimal (19, 9, [1, 1, 1, 1, 1])_6-code:
00452030000000000103 501002004000000003 000010005034020000 300400001000000002 0000400000000320501 010034020000000000 050300001400000000 000000230104000000 400000010020000003 000001002003050004 00200000501000340000 23000000000100400500
7) An Optimal (20, 9, [1, 1, 1, 1, 1])_6-code:
000020000500300040010 5100000034000020000000 0000005040000000132 00000350000000102400 0210004000300000050 00001034002000500000 0400201000003000050 00000102500400300000 04050000003000102000 10000000250000430000 200031000000500000040 03005000000002001004 30200000001000400500 00000000001524000003 000305020040000100000 03420000100050000000
8) An Optimal (22, 9, [1, 1, 1, 1, 1])_6-code:
Lengthening of an optimal (21, 9, [[1, 1, 1, 1, 1]]_6-code.

C. Weight Six Codes
1) An Optimal (20, 11, [3, 3])_3-code:
10000000002021000100 00100200102000020 000221200010000001 0001000202000101100 0100000100002010202
2) An Optimal (20, 11, [3, 2, 1])_3-code:
Refinement of an optimal (20, 11, [3, 3])_3-code.
3) An Optimal (20, 11, [3, 1, 1, 1])_5-code:
Refinement of an optimal (20, 11, [3, 3])_3-code.
4) An Optimal (20, 11, [2, 2, 2])_3-code:
Refinement of an optimal (20, 11, [2, 2])_3-code.
5) An Optimal (21, 11, [2, 2, 2])_4-code:
010000332000100020000 033000000021020000010 302012000300000000001 0001030021000030200 200001020000300030100 00020000100001000323 00020000000213103000
6) An Optimal (22, 11, [2, 2, 2])_4-code:
Lengthening of an optimal (21, 11, [2, 2, 2])_4-code.
7) An Optimal (23, 11, [2, 2, 2])_4-code:
100200200000001000033000 20000031200100030000000 0000000320000200110030 000310200000000001000123 0000000013000013020002 0100000030002300200001 00100000013500002000 02302300000000100000000
8) An Optimal (24, 11, [2, 2, 2])_4-code:
An Optimal


**AUTHORS’ BIOGRAPHIES**

Yeow Meng Chee (SM’08) received the B.Math. degree in computer science and combinatorics and optimization and the M.Math. and Ph.D. degrees in computer science, from the
University of Waterloo, Waterloo, ON, Canada, in 1988, 1989, and 1996, respectively.

Currently, he is an Associate Professor at the Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. Prior to this, he was Program Director of Interactive Digital Media R&D in the Media Development Authority of Singapore, Postdoctoral Fellow at the University of Waterloo and IBM’s Zürich Research Laboratory, General Manager of the Singapore Computer Emergency Response Team, and Deputy Director of Strategic Programs at the Infocomm Development Authority, Singapore. His research interest lies in the interplay between combinatorics and computer science/engineering, particularly combinatorial design theory, coding theory, extremal set systems, and electronic design automation.

Son Hoang Dau received the Bachelors degree in Applied Mathematics and Informatics from the College of Science, Vietnam National University, Hanoi, Vietnam, in 2006 and the M.S. degree in mathematical sciences from the Division of Mathematical Sciences, Nanyang Technological University, Singapore, where he is currently working towards the Ph.D. degree.

His research interests are coding theory and combinatorics.

Alan C. H. Ling was born in Hong Kong in 1973. He received the B.Math., M.Math., and Ph.D. degrees in combinatorics & optimization from the University of Waterloo, Waterloo, ON, Canada, in 1994, 1995, and 1996, respectively. He worked at the Bank of Montreal, Montreal, QC, Canada, and Michigan Technological University, Houghton, prior to his present position as Associate Professor of Computer Science at the University of Vermont, Burlington. His research interests concern combinatorial designs, codes, and applications in computer science.

San Ling received the B.A. degree in mathematics from the University of Cambridge, Cambridge, U.K., in 1985 and the Ph.D. degree in mathematics from the University of California, Berkeley, in 1990.

Since April 2005, he has been a Professor with the Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. Prior to that, he was with the Department of Mathematics, National University of Singapore. His research fields include arithmetic of modular curves and application of number theory to combinatorial designs, coding theory, cryptography and sequences.