New perfect difference families

Tsonka Baicheva Svetlana Topalova

Institute of Mathematics and Informatics
Bulgarian Academy of Sciences

September, 2014
Svetlogorsk, Russia

Motivation I

Construction of high-rate regular quasi-cyclic low-density parity-check (QC LDPC) codes based on a subclass of cyclic difference families

H. Park, S. Hong, J. No and D. Shin, Construction of high-rate regular quasi-cyclic LDPC codes based on cyclic difference families, *IEEE Trans. on Communic.*, **61**, no 8, 3108–3113, 2013.

Motivation II

QC LDPC codes constructed from cyclic difference families (CDFs) are proposed but they have restricted lengths

- S. J. Johnson and S. R. Weller, A family of irregular LDPC codes with low encoding complexity, *IEEE Commun. Lett.*, **7**, no 2, 79–81, 2003.
- B. Vasik and O. Milenkovic, Combinatorial constructions of low-density parity-check codes for iterative decoding, *IEEE Trans. on Inform. Theory*, **50**, no 6, 1156–1176, 2004.
- B. Ammar, B. Honary, Y. Kou, J. Xu and S. Lin, Constructions of low-density parity-check codes based on balanced incomplete block designs, *IEEE Trans. on Inform. Theory*, **50**, no 6, 1257–1268, 2004.
- M. Fujisava and S. Sakata, A construction of high rate quasi-cyclic regular LDPC codes from cyclic difference families with girth 8, *IEICE Trans. Fundamentals*, **E-90A**, no 5, 1055–1061, 2007.

Motivation III

In this work we will focus on PDFs because

- QC LDPC codes with various code rates can be constructed
- QC LDPC codes with various code lengths can be constructed
- Constructed QC LDPC codes can achieve the minimum code length among all possible regular LDPC codes with girth 6 for given parameters, i.e. are useful in designing error-correcting systems which require high rate and short and moderate lengths

Motivation IV

CDFs and PDFs have many other relations and practical applications

- They are related to one-factorizations of complete graphs and to cyclically resolvable cyclic Steiner triple systems
- Very efficient constructions of new optimal perfect secrecy systems that are one-fold secure against spoofing are obtained via CDF
- Optimal frequency-hopping sequences can be constructed from $(v, k, 1)$ CDFs
- They can be used for a construction of other types of combinatorial structures
  - regular perfect systems of difference sets
  - optimal difference triangle sets
  - perfect optimal optical orthogonal codes
  - cyclic 2-(v,k,1) designs, etc

Basic definitions and notations I

R. Julian R. Abel and M. Buratti *Difference families*, in: C.J. Colbourn, J.H. Dinitz (Eds.), *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 270–287, 1996.

- **B** is a subset of an additive group G.
- Δ**B** is the list of all possible differences $b - b'$ with (b, b') an ordered pair of distinct elements of **B**.
- **F** $= \{B_1, B_2, \dots, B_n\}$ is a collection of subsets of G.
- Δ**F** is the list of differences from **F** and is the multiset obtained by joining $\Delta B_1, \dots, \Delta B_n$.

Basic definitions and notations II

Definition

F is said to be a $(v, k, 1)$ difference family (DF) if G has order v, every B_i is of size $k \geq 3$, and $\Delta\mathbf{F}$ covers every non-zero element of G exactly once.

Definition

If $G = Z_v$, then the difference family is said to be cyclic (CDF).

Basic definitions and notations III

- Let F be a CDF
- Denote by $\bar{\Delta}B$ the list of all possible differences $b - b'$ with $b > b'$, where b and b' are distinct elements of B

Definition

If $\bar{\Delta}F = \{1, 2, \ldots, (v-1)/2\}$, then F is called a **perfect difference family**, or briefly, a $(v, k, 1)$ PDF.

- Therefore $(v, k, 1)$ PDFs are a subclass of $(v, k, 1)$ CDFs

Known results for PDFs I

Theorem

1) If $v \equiv 1$ *or* 7 $(mod\ 24)$*, then a* $(v, 3, 1)$ *PDF exists.*

2) A $(12t + 1, 4, 1)$ *PDF exists for* $t = 1$*,* $4 \leq t \leq 1000$*.*

3) $(20t + 1, 5, 1)$ *PDFs are known for* $t = 6, 8, 10$ *but for no other values of* $1 \leq t \leq 50$*.*

4) There are no $(v, k, 1)$ *PDF for the following values:*
a) $k = 3$*,* $v \equiv 13$ *or* 19 $(mod\ 24)$*,*
b) $k = 4$*,* $v \in \{25, 37\}$*,*
c) $k = 5$*,* $v \equiv 21$ $(mod\ 40)$ *or* $v \in \{41, 81\}$*,*
d) $k \geq 6$*.*

Known results for PDFs II

- We know classification results only for $(121, 5, 1)$ PDFs

  Ph. Laufer, Regular perfect systems of difference sets of size 4 and extremal systems of size 3, , *Ann. Discrete Math.*, **12**, 193–201, 1982.

- 4800 PDFs have been obtained

Difference triangles I

- For a k-element set $B = \{b_1, b_2, ...b_k\}$ it is convenient to present the differences from $\bar{\Delta} B$ by a difference triangle D
- Elements of D are $d_i^j = b_{i+j} - b_i$, for $i, j = 1, \ldots, k-1$

Difference triangles for $k = 3, 4$ and 5

k=3	k=4	k=5
		d_1^4
	d_1^3	d_1^3 d_2^3
d_1^2	d_1^2 d_2^2	d_1^2 d_2^2 d_3^2
d_1^1 d_2^1	d_1^1 d_2^1 d_3^1	d_1^1 d_2^1 d_3^1 d_4^1

Difference triangles II

The most important property of a difference triangle is that the sum of the elements in the upper half of the triangle is equal to the sum of the elements in the lower half

k=3	k=4	k=5
d_1^2 d_1^1 d_2^1	d_1^3 d_1^2 d_2^2 d_1^1 d_2^1 d_3^1	d_1^4 d_1^3 d_2^3 d_1^2 d_2^2 d_3^2 d_1^1 d_2^1 d_3^1 d_4^1

$$S_{upper\ \ half}=S_{lower\ \ half}$$

Example I

$(25, 3, 1)$ PDF

$$\mathbf{F} = \{\{0, 1, 6\}\ \{0, 2, 10\}\ \{0, 3, 12\}\ \{0, 4, 11\}\}$$

Coresponing difference triangles

6 10 12 11

1 5 2 8 3 9 4 7

$(49, 4, 1)$ PDF

$$\mathbf{F} = \{\{0, 1, 12, 18\}\ \{0, 2, 7, 22\}\ \{0, 3, 16, 24\}\ \{0, 4, 14, 23\}\}$$

Coresponing difference triangles

18 22 24 23

12 17 7 20 16 21 14 19

1 11 6 2 5 15 3 13 8 4 10 9

Classification algorithm 1 I

- For a $(v, k, 1)$ PDF it holds that $v = k(k-1)m + 1$
- We first construct a list of all possible k-element subsets of the set of the integers from 1 to v, such that their corresponding difference sets do not contain differences which are greater than $k(k-1)m/2$
- We sort the list by the minimum (or maximum) differences of the sets and a lexicographic order defined on the triangles
- We choose the elements of the current PDF by back track search.
- When s sets have been chosen, we add a set containing the smallest (or biggest) difference which is not contained in the already chosen difference triangles.

Classification algorithm 1 II

- We calculate S_{max} - the sum of the biggest $m - s$ (or $3(m - s)$ for $k = 5$) differences which are not contained in the already chosen difference triangles
- For $k = 4$ we calculate S_{min} - the sum of the smallest $(k - 1)(m - s)$ differences which are not contained in the already chosen difference triangles
  - If $S_{max} \geq S_{min}$ we choose an $(s + 1)$-st element
  - If $S_{max} < S_{min}$ we replace the s-th element by the next possible one
- For $k = 3$ and $k = 5$ the sum of the elements of the m first (or m first two) rows of the difference triangles equals S: half of the sum of the first $k(k - 1)m/2$ integers
  - If $S_{max} \geq S$ we choose an $(s + 1)$-st element
  - If $S_{max} < S$ we replace the s-th element by the next possible one

Classification algorithm 1 III

Definition

Two difference families $\mathbf{F} = \{B_1, B_2, \ldots, B_n\}$ and $\mathbf{F}' = \{B'_1, B'_2, \ldots, B'_n\}$ over Z_V are equivalent if there is an automorphism α of Z_V such that for each $i = 1, 2, \ldots, n$ there exists B'_j which is a translate of $\alpha(B_i)$.

- With respect to the defined lexicographic order on the difference triangles, the currently obtained PDF is greater than the previous ones.
- We use this to check each PDF for equivalence to some of those which were constructed earlier by checking if the current solution can be mapped to a lexicographically smaller one by some of the automorphisms of Z_v.
- This way besides the set of all PDFs, we also obtain a set of inequivalent PDFs with the given parameters.

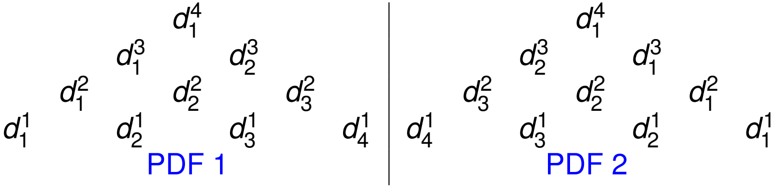
Algorithm 2

- We use a modification of our algorithm for construction of optical orthogonal codes and CDFs
- By this algorithm we classify only the inequivalent PDFs with the given parameters
- Algorithm 1 is much faster, but we use Algorithm 2 to compare part of the obtained results

Implementation

- Our computer implementations of both algorithms are written in C++
- The programmes ran on a PC with an Intel Xeon 2.5 GHz 6 cores processor
- With Algorithm 1 the classification of the (121,5,1) PDFs took 4 days (running in parallel on 12 threads)

Improvement of Algorithm 1



$$\begin{array}{ccccccc} & & & d_1^4 & & & \\ & & d_1^3 & & d_2^3 & & \\ & d_1^2 & & d_2^2 & & d_3^2 & \\ d_1^1 & & d_2^1 & & d_3^1 & & d_4^1 \end{array}$$

PDF 1

$$\begin{array}{ccccccc} & & & d_1^4 & & & \\ & & d_2^3 & & d_1^3 & & \\ & d_3^2 & & d_2^2 & & d_1^2 & \\ d_4^1 & & d_3^1 & & d_2^1 & & d_1^1 \end{array}$$

PDF 2

- We construct only PDFs with $d_1^1 < d_{k-1}^1$ while for each of them $2^m - 1$ other PDFs can be obtained in linear time
- At each step, for each missing difference we count the number of possible sets containing it and we add only sets containing the difference which is in the least number of possible sets
- With the improved Algorithm 1 the classification of the (121,5,1) PDFs took half an hour on the same computer

Results

Table : $(v, k, 1)$ perfect difference families.

v	k	m	ineq. CDFs	PDFs	ineq. PDFs
25	3	4	12	168	12
31	3	5	80	672	68
49	3	8	157340	778240	150788
55	3	9	3027456	10498560	2520064
73	3	12		10567748.2^{12}	
79	3	13		103372655.2^{13}	
13	4	1	1	2	1
49	4	4	224	192	80
61	4	5	18132	5568	2544
73	4	6	1426986	200448	94368
85	4	7		9207040	4552504
97	4	8		2633052.2^{8}	
109	4	9		110905803.2^{9}	
121	5	6		6624	3296
161	5	8		$> 6142.2^{8}$	