BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **18**, No 5
Special Thematic Issue on Optimal Codes and Related Topics
Sofia • 2018 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2018-0016

On Tight Optimal Conflict-Avoiding Codes for 3, 4, 5 and 6 Active Users

Tsonka Baicheva[1,2], *Svetlana Topalova*[1]

[1]*Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria*
[2]*D. A. Tsenov Academy of Economics, 5250 Svishtov, Bulgaria*
E-mails: tsonka@math.bas.bg svetlana@math.bas.bg

Abstract**:** *We classify tight optimal conflict-avoiding codes of weights 3, 4, 5 and 6 and given small lengths.*

Keywords**:** *Conflict-avoiding codes, tight optimal codes, protocol sequences.*

1. Introduction

Conflict-Avoiding Codes (CAC) of weight k, length n and cardinality M can be applied to avoid collisions in channels with asynchronous multiple access without feedback. In that case the number of codewords M corresponds to the maximum number of users of the channel and the weight k to the maximum number of active users at a given moment. It is assumed that time is partitioned into intervals (slots) and all users have slot synchronization. No other synchronization is assumed. The assigned to each user protocol sequences formed from codewords of a suitable CAC, must allow each of k active users to transmit a data packet successfully in one of k attempts during n time slots without collisions with other active users.

Definition 1. A conflict-avoiding code of length n for k active users ((n, k) CAC) is a set $C \subseteq \{(0, 1)\}^n$ of binary vectors, or codewords, all of Hamming weight k, such that arbitrary cyclic shifts x', y' of distinct codewords $x, y \in C$ intersect in at most one coordinate, i.e., $\mathrm{dist}(x', y') \geq 2k - 2$, where $\mathrm{dist}(x', y')$ is the Hamming distance between x' and y'.

Definition 2. The support supp(x) of a codeword x is the set of indices of its nonzero coordinates.

It is more convenient for our investigation to use supp(x) instead of x. Denote supp(x) by X and let $X = \{x_0, x_1, \ldots, x_{k-1}\}$. The support of a cyclic shift of x is then a translate $\{x_0 + t, x_1 + t, \ldots, x_{k-1} + t\}$ of X, where addition is modulo n.

Denote by $\Delta'(X) = \{x_i - x_j \pmod{n} | x_i, x_j \in X, i \neq j\}$ the multiset of differences of X and by $\Delta(X)$ its corresponding set.

Definition 3. The number $|\Delta(X)|$ is called the **type of** X and denoted by $T(X)$.

Definition 4. Any (n, k) conflict-avoiding code C can be considered as a collection of k subsets of Z_n such that

$$\Delta(X) \cap \Delta(Y) = \varnothing \text{ for any } X, Y \in C.$$

Therefore when we talk of codewords below, we will actually mean k subsets of Z_n.

Definition 5. Two codewords are equivalent if $\Delta(X) = \Delta(Y)$.

We assume that $x_0 < x_1 < \ldots < x_{k-1}$ for each codeword $X = \{x_0, x_1, \ldots, x_{k-1}\}$. We define a lexicographic order on the codewords in the following way.

Definition 6. The codeword $X' = \{x'_0, x'_1, \ldots, x'_{k-1}\}$ is lexicographically smaller than $X'' = \{x''_0, x''_1, \ldots, x''_{k-1}\}$ if $T(X') < T(X'')$, or if $|\Delta(X')| = |\Delta(X'')|$ and $x'_i = x''_i$ for $i < j$, but $x'_j < x''_j$ for some j.

Without loss of generality we assume that each codeword is lexicographically smaller than its translates. This means that $x_0 = 0$ for each codeword and when we say that X_1 is mapped to X_2 by the permutation φ, we mean that X_2 is the smallest translate of $\varphi(X_1)$.

Definition 7. An (n, k) CAC of size s is **tight** (**perfect**) if $\bigcup_{i=1}^{s} |\Delta X_i| = n - 1$, that is if all nonzero differences are covered.

Example 1. The four codewords of a tight (15, 3) CAC are listed below together with their sets of differences and types:

$X_1 = \{0, 5, 10\}$	$\Delta(X_1) = \{5, 10\}$	$\mathrm{T}(X_1) = 2$
$X_2 = \{0, 1, 2\}$	$\Delta(X_2) = \{1, 2, 13, 14\}$	$\mathrm{T}(X_2) = 4$
$X_3 = \{0, 7, 11\}$	$\Delta(X_3) = \{4, 7, 8, 11\}$	$\mathrm{T}(X_3) = 4$
$X_4 = \{0, 6, 12\}$	$\Delta(X_4) = \{3, 6, 9, 12\}$	$\mathrm{T}(X_4) = 4$.

Definition 8. Denote by $M(n, k)$ the maximum cardinality of a CAC of length n. A conflict-avoiding code C is said to be **optimal** if $|C| = M(n, k)$.

The advantage of using optimal codes is that they enable the largest number of asynchronous users to transmit packets efficiently and reliably through a multiple-access channel without feedback.

Definition 9. Two (n, k) CACs are multiplier equivalent if they can be obtained from one another by a multiplier automorphism of Z_n and replacement of codewords by some of their translates.

Remark. Any CAC of length n for k active users can be viewed as an $(n, k, k, 1)$ optical orthogonal code. Such codes are used in optical code-division multiple access channels [1].

Optimal CACs as protocol sequences for a multiple-access collision channel without feedback have been studied in many works [2-18]. The case with three active users ($k = 3$) is completely settled. Several optimal constructions for weights 4 and 5 can be found in [14]. General upper bounds on the size of constant weight CACs applicable to all code lengths and all Hamming weights are derived in [15] and [16]. Examples of small length CACs can be found in [17], and classification results about CACs of weights up to 7 and small lengths in [2] and [3].

Tight CACs have additional properties which make them interesting as incidence structures and thus might be more appropriate in constructions of other codes or combinatorial structures. There are, for instance, recursive constructions of CACs from tight CACs of smaller lengths [11]. That is why tight CACs are of particular importance.

Some existence conditions and constructions of tight $(n, 3)$ CACs for definite values of n can be found in [13, 5, 18, 10, 11], but checking the conditions and applying the constructions is often a difficult job for those who are interested in such codes from any application point of view. There are tight CACs among those which are classified in [2, 3], but they are not explicitly listed. That is why we think that the online availability of the nonequivalent tight CACs of small k and n will be useful for any application purposes. This motivated us to present the current classification of tight optimal CACs.

Our investigation concerns optimal tight CACs of weight $k = 3$ and length $n \leq 111$, $k = 4$ and $n \leq 120$, $k = 5$ and $n \leq 118$, $k = 6$ and $n \leq 119$, and $k = 7$ and $n \leq 95$. The construction algorithm which we use here, allows us to classify the tight codes for 21 lengths for which the optimal CACs are not classified in [2, 3].

2. Algorithm

Our algorithm constructs tight codes with a given cardinality s. It performs backtrack search on the set of all *possible codewords*. We obtain them in advance from all nonequivalent (by Definition 5) k-sets of Z_n. We find the type of the codewords defined by these k-sets. Let $T_{\min}$ be the smallest type. We want to construct a code of s codewords. Such a code cannot have codewords of a type greater than $T_{\max} = n - 1 - (s - 1)T_{\min}$. Possible codewords are only k-sets of type at most $T_{\max}$. Therefore the set of possible codewords for a code of cardinality $s + 1$ might be much smaller than that of a code of cardinality s.

The possible codewords are partitioned in groups, such that each multiplier automorphism of Z_n maps any codeword to a codeword of the same group. We call leader the lexicographically smallest codeword of the group. We sort the groups with respect to the lexicographic order of their leaders and save them. For each possible codeword we know the possible codeword to which it is mapped by any automorphism of Z_n, and this makes the below described minimality test very fast.

The set of possible codewords we construct here differs from the similar sets used in [2] and [3]. In [2] equivalence of codewords is not defined, while here we give Definition 5 following [18]. By this definition we filter away possible codewords which might lead to codes which are different as combinatorial structures, but which perform the same as CACs for channels with asynchronous multiple access without feedback. The smaller search set makes the classification algorithm much faster. In [3] we remove a group from the set of possible codewords if the set of differences of its leader is the same as the set of differences of another leader, while here we remove a group from the set of possible codewords if the set of differences of its leader is the same as the set of differences of any possible codeword from another group.

The backtrack search on the possible codewords is similar to the one we used in [2] and [3], namely when we add the next codeword, we speed up the algorithm by performing a Minimality test and a Type test to the current partial solution. Apart from this, to construct tight codes here, we apply a Tight test too. We briefly describe the three tests.

Suppose that r codewords of the code have been already found. Let T be the type of the r-th codeword, and let d be the number of distinct differences covered by the r codewords.

Type test. We only look for codes with a definite number s of codewords. That is why the type of the remaining possible codewords (of the array we choose them from) is at least as big as that of the r-th chosen one. That is why $d + (s - r)T \leq n - 1$. If this does not hold, the next possibility for the $(r-1)$-st codeword is considered.

Tight test. Knowing the types of the remaining possible codewords, we try to find codeword types $T_1, \ldots, T_{s-r}$ such that $d + T_1 + \ldots + T_{s-r} = n - 1$. If this does not hold, the next possibility for the $(r - 1)$-st codeword is considered.

Minimality test. We check if the current partial solution can be mapped to a lexicographically smaller one by some of the automorphisms of Z_n. If it can, an equivalent partial solution has already been considered, and we look for the next possibility for the current codeword.

3. Results

We consider values of n for which the cardinality of the optimal CACs is known [2, 3]. The results are summarized in the tables below. Only tight CACs with at least 2 codewords are included. Lengths for which no tight optimal CAC exists, are not presented. The number of nonequivalent tight optimal CACs is given in column TCACs. One can see that this number is very big if the size of the optimal codes is relatively small for the given length. Tight optimal codes with relatively big sizes for the given length range are most interesting and usually there are not many of them.

All the constructed codes are available online and can be downloaded from **http://www.moi.math.bas.bg/~svetlana**. Information on the different types of codes (with respect to the types of the codewords) is also given there as illustrated in the following example.

Example 2. There are nine nonequivalent (119, 6) tight optimal CACs of four code-types which are presented as:

0) 2:	6-1	10-5	12-3	26-1
1) 2:	6-1	10-4	12-4	24-1
2) 1:	6-1	12-8	16-1	
3) 4:	6-1	12-7	14-2	

This means that there are two codes of code-type 0 (with one codeword of type 6, five codewords of type 10, three of type 12 and one of type 26), two codes of code-type 1 , one of code-type 2 and four of code-type 3.

Table 1. Tight optimal CACs with $k = 3$ and $n \le 111$

n	$M(n, 3)$	TCACs
13	3	1
15	4	1
16	3	2
17	4	1
18	4	2
19	4	2
20	4	3
21	4	5
23	5	1
24	5	8
25	6	1
27	6	2
28	6	5
29	7	1
30	7	2
31	7	2
32	7	2
33	8	3
35	8	1
36	8	30
37	9	1
39	10	2
40	8	195
41	10	1
42	10	1
43	10	2
44	9	308
47	11	1
48	10	1,602
49	11	22
51	13	4
52	11	621
53	13	1
54	13	2
56	12	170
57	14	5
59	14	2
60	13	7,702
61	15	1
63	15	46
64	13	101,136
65	16	6
66	16	1
67	16	4
68	15	200
69	17	7
71	17	3
72	16	1,333
73	17	80
75	19	4
76	16	377,203
77	18	78
78	19	1
79	19	5
80	17	209,575
81	19	1,758
83	20	4
84	19	2,464
85	21	10
87	22	2
88	19	39,552
89	21	125
90	22	3
91	22	4
92	20	200,224
93	23	6
95	23	2
96	21	3,411,597
97	24	1
99	24	40
100	22	40,928
101	25	1
102	25	1
103	25	5
104	22	≥5,000,000
105	26	22
107	26	2
108	24	≥3,600,000
109	27	2
111	28	2

Table 2. Tight optimal CACs with $k = 4$ and $n \le 120$

n	$M(n, 4)$	TCACs
17	2	1
20	3	2
24	3	2
25	3	1
28	4	4
30	4	2
32	5	1
34	5	1
35	6	1
36	5	13
37	6	1
38	6	2
39	5	1
40	6	24
41	6	2
42	6	3
43	6	11
44	7	6
45	6	66
46	7	6
47	7	1
57	8	3
58	9	1
59	9	1
60	9	57
61	9	4
62	9	302
63	9	2
64	10	6
65	10	5
67	10	4
68	10	629
69	11	0
70	11	9
71	10	440
72	11	11
73	11	3
74	11	294
75	11	38
76	12	5
77	12	1
78	11	26,106
88	14	4
89	13	517
90	14	8
91	14	1
92	14	572
94	14	3,827
95	14	4,386
96	14	22,577
97	15	2
98	15	72
99	14	160,321
100	16	21
101	15	337
102	15	17,812
104	17	2
105	16	11
106	16	2,950
107	16	245
108	16	10,812
109	16	6,676
110	17	692

Table 2 (continued)

n	*M*(*n*, 4)	TCACs
48	7	24
49	8	1
50	7	350
51	7	4
52	8	18
53	7	225
55	8	37
56	8	310
79	12	3
80	13	2
82	12	3,797
83	12	470
84	13	1
85	13	57
86	13	206
87	13	2
112	17	4,858
113	17	231
114	17	9,480
115	17	95,927
116	18	340
117	17	162,282
118	18	2,455
120	18	45,890

Table 3. Tight optimal CACs with $k = 5$ and $n \le 118$

n	*M*(*n*, 5)	TCACs
43	3	1
45	6	1
47	5	1
50	6	1
54	6	1
55	6	7
56	6	2
60	6	23
61	6	10
62	6	5
63	8	2
64	6	15
65	7	11
66	7	4
67	7	2
69	8	1
70	8	8
71	7	13
72	7	64
73	8	15
75	8	34
77	8	362
78	8	47
79	8	19
80	8	59
81	9	4
82	8	39
83	8	285
84	8	1,302
85	9	88
86	9	1
88	9	2
89	9	84
91	10	6
93	10	9
95	11	13
96	9	4,214
97	12	2
98	10	25
99	12	2
101	10	468
102	10	1,942
103	10	2,104
104	10	2,028
105	11	1,521
107	11	67
108	11	58
109	11	372
110	12	18
111	12	94
112	11	1,064
113	12	6
114	12	13
115	13	44
117	14	2
118	12	239

Table 4. Tight optimal CACs with $k = 6$ and $n \le 119$

n	*M*(*n*, 6)	TCACs
42	3	1
62	4	1
66	5	1
69	5	1
72	5	1
77	8	1
84	8	5
88	7	2
89	6	2
90	7	1
91	8	20
94	8	1
96	8	2
98	8	45
99	8	9
100	8	3
103	8	4
104	8	13
107	8	38
108	9	2
111	9	1
112	9	6
116	9	4
117	9	60
119	10	9

Table 5. Tight optimal CACs with $k = 7$ and $n \le 95$

n	*M*(*n*, 7)	TCACs
60	2	1
63	4	2
91	8	1

Acknowledgements**:** The research of the first author was partially supported by the Bulgarian National Science Fund, Contract No DH 12/8, 15.12.2017. The research of the second author was partially supported by the Bulgarian National Science Fund, Contract No DH 02/2, 13.12.2016.

References

1. Chung, F. R. K., J. A. Salehi, V. K. Wei. Optical Orthogonal Codes: Design, Analysis, and Applications. – IEEE Trans. Inform. Theory, Vol. **35**, 1989, No 3, pp. 595-604.
2. Baicheva, T., S. Topalova. Optimal Conflict-Avoiding Codes for 3, 4 and 5 Active Users. – Problems of Information Transmission, Vol. **53**, 2017, No 1, pp. 42-50.
3. Baicheva, T., S. Topalova. Classification of Optimal Conflict-Avoiding Codes of Weights 6 and 7. – Electronic Notes in Discrete Mathematics, Vol. **57**, March 2017, pp. 9-14.
4. Fu, H., Y. Lin, M. Mishima. Optimal Conflict-Avoiding Codes of Even Length and Weight 3. – IEEE Trans. Inform. Theory, Vol. **56**, 2010, No 11, pp. 5747-5756.
5. Fu, H., Y. Lo, K. Shum. Optimal Conflict-Avoiding Codes of Odd Length and Weight Three. – Des. Codes Cryptogr., Vol. **72**, 2014, No 2, pp. 289-309.
6. Jimbo, M., M. Mishima, S. Janiszewski, A. Y. Teymorian, V. D. Tonchev. On Conflict-Avoiding Codes of Length $n = 4m$ for Three Active Users. – IEEE Trans. Inform. Theory, Vol. **53**, 2007, No 8, pp. 2732-2742.
7. Levenshtein, V. I. Conflict-Avoiding Codes for Many Active Users. – In: Abstarcts of 14th Internat. Conference Problems of Theoretic Cybernetics, Penza, 2005, pp. 86-86 (in Russian).
8. Levenshtein, V. I., V. D. Tonchev. Optimal Conflict-Avoiding Codes for Three Active Users. – In: Proc. of IEEE Internat. Symposium on Inform. Theory, Adelaide, 2005, pp. 535-537.
9. Levenshtein, V. I. Conflict-Avoiding Codes and Cyclic Triple Systems. – Probl. of Inform. Transm., Vol. **43**, 2007, No 3, pp. 199-212.
10. Lin, Y., M. Mishima, J. Satoh, M. Jimbo. Optimal Equi-Difference Conflict-Avoiding Codes of Odd Length and Weight Three. – Finite Fields Appl., Vol. **26**, 2014, pp. 49-68.
11. Ma, W., C. Zhao, D. Shen. New Optimal Constructions of Conflict-Avoiding Codes of Odd Length and Weight 3. – Des. Codes Cryptogr., Vol. **73**, 2014, No 3, pp. 791-804.
12. Mishima, M., H. Fu, S. Uruno. Optimal Conflict-Avoiding Codes of Length $n \equiv 0 (\mathrm{mod}\ 16)$ and Weight 3. – Des. Codes Cryptogr., Vol. **52**, 2009, No 3, pp. 275-291.
13. Momihara, K. Necessary and Sufficient Conditions for Tight Equi-Difference Conflict Avoiding Codes of Weight 3. – Des. Codes Cryptogr, Vol. **45**, 2007, No 3, pp. 379-390.
14. Momihara, K., M. Müler, J. Saton, M. Jimbo. Constant Weight Conflict-Avoiding Codes. – SIAM J. Discr. Math., Vol. **21**, 2007, No 4, pp. 959-979.
15. Shum, K. W., W. S. Wong. A Tight Asymptotic Bound on the Size of Constant-Weight Conflict-Avoiding Codes. – Des. Codes Cryptogr., Vol. **57**, 2010, No 1, pp. 1-14.
16. Shum, K. W., W. S. Wong, C. S. Chen. A General Upper Bound on the Size of Constant-Weight Conflict Avoiding Codes. – IEEE Trans. Inform. Theory, Vol. **56**, 2010, No 7, pp. 3265-3276.
17. Tonchev, V. D. Tables of Conflict-Avoiding Codes. **http://www.math.mtu.edu/ tonchev/CAC.html**
18. Wu, S. L., H. L. Fu. Optimal Tight Equi-Difference Conflict-Avoiding Codes of Length $n = 2^k \pm 1$ and Weight 3. – J. Comb. Des., Vol. **2**1, 2013, No 6, pp. 223-231.

Received 30.09.2017; Accepted 07.12.2017