Optimal $(v, 4, 2, 1)$ optical orthogonal codes with small parameters

Tsonka Baicheva, Svetlana Topalova ,

Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Bulgaria

Abstract

Optimal $(v, 4, 2, 1)$ optical orthogonal codes (OOC) with $v \leq 75$ and $v \neq 71$ are classified up to isomorphism. One $(v, 4, 2, 1)$ OOC is presented for all $v \leq 181$, for which an optimal OOC exists.

Keywords: optical orthogonal code; classification; automorphisms of the cyclic group;

1 Introduction

Optical orthogonal codes, initially proposed for application in optical code-division multiple-access communication systems, receive increasing interest by both the research and industrial communities. This is mainly due to the ability to implement data transmission at ultra-high rates. The use of optical orthogonal codes enables a large number of asynchronous users to transmit information efficiently and reliably. The lack of a network synchronization requirement enhances the flexibility of the system. But these codes can also be used in other wide-band code-division multiple-access environments. Optical orthogonal codes are also called cyclically permutable constant weight codes in connection to constructing protocol sequences for multiuser collision channel without feedback.

So far a number of families of optical orthogonal codes have been constructed, see for instance [3, 4, 5, 10, 12, 15]. In this work we not only construct new optimal optical orthogonal codes with $v <= 181$, but also classify all optimal $(v, 4, 2, 1)$ optical orthogonal codes with $v <= 75$ and $v \neq 71$.

2 Preliminaries

For the basic concepts and notations concerning optical orthogonal codes and related designs we follow [3], [4], and [6]. We denote by Z_v the ring of integers modulo v.

A $(v, k, \lambda_a, \lambda_c)$ optical orthogonal code (OOC) can be defined as a collection $\mathcal{C} = \{C_1, ..., C_s\}$ of k-subsets (*codeword-sets*) of Z_v such that any two distinct translates of a codeword-set share at most λ_a elements while any two translates of two distinct codeword-sets share at most λ_c elements:

$$|C_i \cap (C_i + t)| \leq \lambda_a, \quad 1 \leq i \leq s, \quad 1 \leq t \leq v - 1 \tag{1}$$

$$|C_i \cap (C_j + t)| \leq \lambda_c, \quad 1 \leq i < j \leq s, \quad 0 \leq t \leq v - 1 \tag{2}$$

Condition (1) is called the auto-correlation property and (2) the cross-correlation property. The size of $\mathcal{C}$ is the number s of its codeword-sets. A (v, k, λ, λ) OOC is also denoted by (v, k, λ) OOC.

Consider a codeword-set $C = \{c_1, c_2, \ldots, c_k\}$. Denote by $\triangle' C$ the multiset of the values of the differences $c_i - c_j$, $i \neq j$, $i, j = 1, 2, \ldots, k$. The autocorrelation property means that at most λ_a differences are the same. Denote by $\triangle C$ the underlying set of $\triangle' C$. The type of C is the number of elements of $\triangle C$, i.e. the number of different values of its differences. If $\lambda_c = 1$ the cross-correlation property means that $\Delta C_1 \bigcap \Delta C_2 = \emptyset$ for two codeword-sets C_1 and C_2 of the $(v, k, \lambda_a, 1)$ OOC.

Let $V = \{P_i\}_{i=1}^{v}$ be a finite set of *points*, and $\mathcal{B} = \{B_j\}_{j=1}^{b}$ a finite collection of k-element subsets of V, called *blocks*. $D = (V, \mathcal{B})$ is a *design (partial design)* with parameters t-(v,k,λ) if any t-subset of V is contained in exactly (at most) λ blocks of $\mathcal{B}$. Partial designs are also known as *packings* [14] or *packing designs* [10]. We call them partial designs following [4].

A t-(v,k,λ) design (partial design) is *cyclic* if it has an automorphism α permuting its points in one cycle, and it is *strictly cyclic* if each block orbit under this automorphism is of length v (no short orbits). When we talk of block orbit hereafter, we mean block orbit under the automorphism α permuting the points in one cycle.

A circulant matrix of order v is a (0,1) square matrix $M = (m_{i,j})_{v \times v}$ with v rows and columns, such that $m_{i+1,j+1} = m_{i,j}$, where $i, j = 0, 1, ..., v - 1$ and indexes are added modulo v. The incidence matrix of a strictly cyclic design contains circulant matrices of order v, which correspond to the block orbits.

From the $(v, k, \lambda_a, \lambda_c)$ OOC $\mathcal{C}$ one can construct a cyclic t-(v,k,λ) partial design D, which has as blocks the codeword-sets of $\mathcal{C}$ and their translates. Each codeword-set and its translates form one block orbit. The OOC can be reconstructed from this partial design by chosing for codeword-sets exactly one block from each block orbit. In particular, the partial design related to a $(v, 4, 2, 1)$ OOC is a partial 2-(v,4,2) design and at the same time a partial 3-(v,4,1) design. An example is provided in Figure 1., where in a) the two codeword-sets of an optimal (20,4,2,1) OOC are presented. The related strictly cyclic partial 2-(20,4,2) design is given in b). Its incidence matrix contains submatrices, which are circulant matrices of order 20. Each such circulant matrix corresponds to the collection of one of the codewords and its translates.

Figure 1: Example 1. OOC and partial design

a) Optimal perfect (20,4,2,1) OOC $\mathcal{C}$

codeword-sets	differences	distinct differences	type
{0,1,5,6}	1 19 5 15 4 16 6 14 5 15 1 19	1 4 5 6 14 15 16 19	8
{0,2,9,12}	2 18 9 11 7 13 12 8 10 10 3 17	2 3 7 8 9 10 11 12 13 17 18	11

b) Related strictly cyclic partial 2-(20,4,2) design

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
3	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
4	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
5	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
6	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1
2	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
3	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
4	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
5	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0
6	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
7	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0
8	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1
9	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0
10	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0
11	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1
12	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
13	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
14	0	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0
15	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0
16	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0
17	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0
18	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0
19	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	1

This shows that OOCs can also be treated as cyclic combinatorial objects. The automorphisms of the cyclic group of order v map each circulant matrix of order v to a circulant matrix of order v. That is why *multiplier equivalence* [8], [9] is defined for cyclic combinatorial objects. It can be defined for OOCs too.

Definition 1 *Two $(v, k, \lambda_a, \lambda_c)$ optical orthogonal codes C and C' are isomorphic if there exists a permutation of Z_v, which maps the collection of translates of each codeword-set of C to the collection of translates of a codeword-set of C'.*

Definition 2 *Two $(v, k, \lambda_a, \lambda_c)$ optical orthogonal codes are multiplier equivalent if they can be obtained from one another by an automorphism of Z_v and replacement of codeword-sets by some of their translates.*

Two OOCs can be isomorphic, but multiplier inequivalent.

Optical orthogonal codes have various applications [2], [5], [7]. In particular (v, k, λ) OOCs have been widely studied, especially $(v, k, 1)$ and $(v, k, 2)$ OOCs. A long list of publications on them is presented in [3]. OOCs with parameters $(v, 4, 2, 1)$ were first considered in [15]. Recently it was proved in [12] that if

s is the size of a $(v, 4, 2, 1)$ OOC, then

$$s \leq \lceil v/8 \rceil \quad if \quad v \equiv 7, 14 \ (mod\ 56) \tag{3}$$

$$s \leq \lfloor v/8 \rfloor \quad otherwise. \tag{4}$$

A $(v, 4, 2, 1)$ OOC is *optimal* if it reaches this upper bound.

A lot of constructions of infinite families of optimal $(v, 4, 2, 1)$ OOCs and some nonexistence results are presented in [3] and [12]. Yet there are still plenty of values of v, for which it is not known whether such an OOC exists or not. In the present paper we answer this question for all undecided $v < 182$, and we classify up to isomorphism optimal $(v, 4, 2, 1)$ OOCs with $v < 76$. The proofs in [3] show that for some infinite families the existence of optimal OOCs with the smallest parameters is sometimes more difficult to prove theoretically and computer search is suitable then (see for instance [3], Theorem 4.6, Theorem 6.3). And for some families OOCs with additional properties are needed and classification results would be useful. For instance the remark after Theorem 4.7 of [3] shows that this theorem might be more general if there exists an (88,4,2,1) OOC which has one codeword-set whose differences are precisely the non-zero elements of the subgroup of order 8 of Z_{88} (see the first codeword-set of the OOC with $v = 88$ in Table 2). In this sense both existence and classification results for OOCs of small orders might contribute to future investigations on big orders.

A table of optimal $(v, 4, 2)$ OOCs with $v \leq 44$ (with 3 possible exceptions) is presented in [4], where the authors construct them using an algorithm based on the maximum clique search problem. Our approach is essentially different since our aim is not only to find one optimal OOC for each v, but to make a classification too. The classification up to isomorphism of cyclic designs with some parameters [8], [9] was done by first making a classification up to multiplier equivalence. This is the way we proceed here too. Using the same approach, in [1] we classify up to isomorphism $(v, 4, 1)$ OOCs and 2-(v,4,1) designs with $v \leq 76$. The present classification up to isomorphism is, however, more difficult to make than that in [1], because now some of the collections of translates of codeword-set vectors are multiplier inequivalent but isomorphic. The classification up to multiplier equivalence is also more complicated, because in [1] all codeword-sets are of one and the same type, while here the type of the codeword-sets has to be taken in consideration too.

3 Classification up to multiplier equivalence

We classify the $(v, 4, 2, 1)$ OOCs up to multiplier equivalence applying the well-known techniques of back-track search with minimality test on the partial solutions [11, section 7.1.2]. We first arrange all possibilities for codeword-sets with respect to a lexicographic order defined on them.

We assume that $c_1 < c_2 < c_3 < c_4$ for each codeword-set $C = \{c_1, c_2, c_3, c_4\}$. Define a lexicographic order on the codeword-sets implying that: $C' = \{c'_1, c'_2, c'_3, c'_4\}$ is lexicographically smaller than $C'' = \{c''_1, c''_2, c''_3, c''_4\}$ if the type of C' is smaller than that of C'', or if the types of the two codewords are the same and $c'_i = c''_i$ for $i < a$ and $c'_a < c''_a$. If we replace a codeword-set $C \in \mathcal{C}$ with a translate $C + t \in \mathcal{C}$, we obtain an equivalent OOC. That is why without loss of generality we assume that each codeword-set vector of the optimal $(v, 4, 2, 1)$ OOCs is lexicographically smaller than the codeword-set vectors of its translates. This obviously means that $c_1 = 0$.

Let $\varphi_0, \varphi_1, ...\varphi_{m-1}$ be the automorphisms of Z_v, where φ_0 is the identity. We construct an array of all sets of 4 elements of Z_v which might become codeword-set vectors, i.e. which answer the autocorrelation property and are smaller than all their translates. We find them in lexicographic order. To each constructed set we apply the permutations $\varphi_i, i = 1, 2, ...m - 1$. If some of them maps it to a smaller set, we do not add the current set since it is already somewhere in the array. If we add the current set to the array, we also add after it the $m - 1$ sets to which it is mapped by $\varphi_1, \varphi_2, ...\varphi_{m-1}$.

We then apply back-track search to choose the codeword-sets of the OOC among all these possibilities for them. The above described ordering of all the possible codeword-sets allows repeated sets in the array, but makes the minimality test of the partial solutions very fast. By the minimality test we check if the current solution can be mapped to a lexicographically smaller one by the automorphisms of Z_v.

We also apply a type test to the partial solutions. Suppose we have already found r codeword-sets of the code. Let T be the type of the r-th codeword-set, and let d be the number of distinct differences covered by the r sets. We only look for optimal codes, i.e. codes with s codeword-sets. The type of the remaining codeword-sets (of the array we choose them from) is at least as big as that of the r-th chosen one. That is why $d + (s - r)T \leq v - 1$. If this does not hold, we look for the next possibility for the $r - 1$-st codeword-set.

In this way we classify the OOCs up to multiplier equivalence.

4 Isomorphism test

We first apply an isomorphism test to the possible codeword-sets we use. To each set we relate a circulant made of its translates.

Theorem 4.1 *For all $(v, 4, 2, 1)$ OOCs with $16 \leq v \leq 75$ and $v \neq 0$ (mod 8) any two circulants related to two possible codeword-sets are isomorphic iff they are multiplier equivalent.*

Proof. We check this by computer. For most values of v we use our software for establishing design isomorphism. For some values of v, however, the test for isomorphism of the circulants is a difficult task for this general case software. That is why for some v we use the set of generalized multipliers, defined by Muzychuk [13]. This is a set of permutations of Z_v, which is defined for each value of v. Muzychuk [13] proves that two circulants of order v are isomorphic iff they can be mapped to one another by some of these generalized multipliers.

$\diamondsuit$

For all $16 \leq v \leq 75$ and $v = 0$ (mod 8), however, there exist codeword-sets, which are multiplier inequivalent, but isomorphic.

Theorem 4.2 *Any two multiplier inequivalent $(v, 4, 2, 1)$ OOCs with $16 \leq v \leq 75$ are non isomorphic.*

Proof. For $(v, 4, 2, 1)$ OOCs with $16 \leq v \leq 75$ and $v \neq 0$ (mod 8) this follows from Theorem 4.1 and the definition of isomorphic OOCs.

For $v = 16, 24, 32, 40, 48, 56, 64$ and 72 we check this by computer. We test for possible isomorphism all the OOCs, containing at least one codeword-set which is multiplier inequivalent, but isomorphic to some other codeword-set of this or of another OOC. We check for each OOC whether there is a permutation which transforms it into another multiplier inequivalent OOC. We try all permutations of the set of generalized multipliers defined by Muzychuk [13]. No isomorphic and multiplier inequivalent OOCs are found.

5 Classification and existence results

Files with all (v,4,2,1) OOCs we construct can be downloaded from http://www.moi.math.bas.bg/∼ tsonka. We present in Table 1 the results of the classification up to isomorphism of optimal (v,4,2,1) OOCs with $v \leq 75$. For values of v, for which there is no optimal OOC with these parameters, we classify OOCs with one codeword-set less than the optimal one. We call such codes *best*. That is why in the column s_O the number of codeword-sets of the optimal code is presented. The value there is the same as that in the next column s if the classified codes are optimal. We present the codeword-sets of one of the OOCs with this v. All codeword-sets contain 0, which we do not write to save place. If there are perfect ones, the OOC we present in the table is perfect. In Tables 2, 3, 4 and 5 one OOC is given for each $v <= 181$. The sign $\surd$ stands in column p if this OOC is perfect.

Table 1: Classification of optimal (v,4,2,1) OOCs with $v \leq 75$

v	s_O	s	all	perfect	codeword-sets of one OOC								
16	2	1	20	0	1 2 9								
17	2	2	1	1	1 4 5	2 8 10							
18	2	1	30	0	1 2 10								
19	2	2	1	0	1 4 5	2 8 10							
20	2	2	10	1	1 5 6	2 10 13							
21	2	2	7	0	1 3 19	4 10 14							
22	2	2	23	0	1 2 12	3 8 17							
23	2	2	19	0	1 2 4	5 11 16							
24	3	2	113	0	1 2 13	4 9 14							
25	3	3	1	1	1 4 22	2 10 12	5 11 16						
26	3	3	5	0	1 2 14	3 7 10	5 11 20						
27	3	2	192	0	1 2 4	5 10 20							
28	3	3	44	5	2 8 22	3 12 19	1 11 24						
29	3	3	21	0	1 2 4	5 10 17	6 14 20						
30	3	3	156	0	1 2 16	3 7 10	5 11 24						
31	3	3	119	0	1 2 4	5 10 18	6 15 22						
32	4	3	642	2	2 8 10	1 4 17	5 14 25						
33	4	3	585	4	1 4 30	5 13 24	2 12 18						
34	4	4	21	4	1 9 26	3 15 22	4 14 28	2 13 18					
35	4	4	28	7	1 2 4	6 12 24	7 15 22	5 10 26					
36	4	4	72	18	1 17 18	7 15 28	4 14 26	2 27 33					
37	4	4	155	13	1 3 35	7 17 24	6 15 21	4 12 23					
38	4	4	1467	68	1 2 20	4 13 17	3 6 30	5 15 31					
39	4	4	797	86	1 2 4	5 14 19	6 12 22	7 15 28					
40	5	5	11	11	1 2 21	3 6 12	4 14 18	5 16 29	7 15 32				
41	5	5	3	3	1 3 39	4 14 31	6 19 25	7 15 33	9 20 29				
42	5	5	139	26	6 12 24	1 2 22	3 7 38	8 16 25	5 15 28				
43	5	5	107	30	1 2 4	7 19 26	6 21 27	9 20 29	5 10 35				
44	5	5	1938	377	1 2 23	5 10 20	8 19 27	6 18 32	3 31 40				
45	5	5	624	158	1 2 4	8 16 32	6 20 31	9 19 28	5 12 27				
46	5	5	16962	1550	1 2 24	3 6 12	8 16 35	5 20 33	4 29 36				
47	5	5	12214	1277	1 2 4	5 10 20	9 21 30	6 25 39	7 18 31				
48	6	5	113629	4674	1 2 25	7 14 28	6 19 38	4 22 37	3 8 39				
49	6	6	96	82	7 14 28	1 2 4	5 18 31	6 22 33	8 20 37	9 24 39			
50	6	6	2447	1130	1 2 26	3 6 12	4 19 23	5 16 21	7 20 37	8 18 36			
51	6	5	514731	10504	1 2 4	5 10 20	8 29 38	6 18 32	7 24 35				
52	6	6	18853	4748	1 2 27	3 6 12	5 19 33	10 20 41	4 22 39	7 36 44			
53	6	6	15030	4020	1 2 4	5 10 20	6 12 24	7 16 23	11 22 39	8 27 40			
54	6	6	214583	25000	1 2 28	5 10 20	3 19 35	9 23 40	4 12 33	6 13 24			
55	6	6	169834	23774	1 2 4	6 12 24	13 27 40	10 21 44	5 30 38	7 26 46			
56	7	7	811	663	8 16 32	1 2 29	3 12 15	7 18 25	5 19 42	6 26 36	4 17 39		
57	7	6	1279381	79334	1 2 4	5 10 20	8 16 32	7 18 46	9 23 35	6 19 36			
58	7	7	10271	3903	1 2 30	3 6 12	11 22 44	7 26 39	5 20 43	8 24 42	4 31 41		
59	7	7	7932	3291	1 2 4	5 10 20	7 14 28	6 24 30	9 26 42	8 27 40	11 22 47		
60	7	7	289139	60646	1 2 31	7 14 28	11 22 44	8 18 26	3 6 12	5 20 25	4 23 47		
61	7	7	122586	29927	1 2 4	5 10 20	7 14 28	12 24 48	8 19 50	6 29 45	9 26 43		
62	7	7	1672477	193886	1 2 32	3 6 12	7 14 28	10 23 36	5 25 42	8 24 43	4 22 51		
63	8	8	2823	2823	9 18 36	1 2 4	5 10 20	6 23 29	7 14 28	8 16 32	11 22 41	12 25 37	
64	8	8	2354	2354	1 2 33	3 6 12	4 14 18	5 24 45	7 27 34	8 23 49	11 28 39	13 29 42	
65	8	8	2610	2610	1 2 4	5 10 20	6 19 25	7 28 35	8 26 34	9 23 32	11 22 38	12 24 41	
66	8	8	238215	89236	1 2 34	5 10 20	7 14 28	4 8 39	9 26 49	3 16 53	6 12 24	11 22 47	
67	8	8	59871	24721	1 2 4	5 10 20	7 14 28	11 22 44	8 27 35	13 29 42	9 26 50	6 37 55	
68	8	8	1364771	304507	1 2 35	3 6 12	5 10 20	7 14 28	11 27 52	8 31 39	13 30 49	4 26 50	
69	8	8	2365589	540319	1 2 4	5 10 20	7 14 28	11 22 44	9 26 43	6 18 57	8 27 46	13 37 53	
70	9	9	1417	1417	10 20 40	1 2 36	3 6 12	4 18 56	5 26 49	7 22 55	8 27 51	11 28 39	13 29 54
71	8	8	$\geq$6116889	$\geq$744092	1 8 9	2 16 18	3 24 27	6 17 60	12 32 51	4 33 37	13 26 48	5 15 46	
72	9	9	86028	86028	1 2 37	3 6 12	4 8 38	5 28 49	7 20 27	10 29 53	11 33 50	14 32 46	15 31 56
73	9	9	17021	17021	1 2 4	5 43 48	12 27 39	23 40 56	24 42 55	26 45 54	37 44 66	41 52 62	53 59 67
74	9	9	740033	287113	1 12 13	9 34 43	8 27 55	3 20 23	7 22 29	2 35 41	4 30 48	10 24 60	5 37 58
75	9	9	1944427	774968	1 11 65	2 22 55	5 13 18	4 27 52	7 31 38	15 34 49	12 28 40	6 36 45	3 32 61

Table 2: Examples of best or optimal (v,4,2,1) OOC

v	s_O	s	p	codeword-sets								
76	9	9		1 2 39	3 24 27	4 22 58	9 32 41	10 15 20	11 19 30	13 29 60	14 42 48	17 43 50
77	9	9		11 22 44	1 52 53	2 15 64	3 9 74	4 14 18	5 32 37	7 36 43	8 38 46	16 35 58
78	9	9		1 2 40	3 20 23	4 47 51	6 24 30	7 12 19	8 21 29	9 25 34	10 32 42	11 26 37
79	9	9		1 2 4	5 24 29	6 21 64	7 20 27	8 17 25	10 28 38	11 37 48	12 34 46	14 30 44
80	10	10	$\surd$	1 2 41	3 6 74	4 26 58	5 23 28	13 34 59	14 24 38	15 30 50	16 33 49	29 36 73
				53 61 72								
81	10	10	$\surd$	1 2 4	5 19 24	6 39 45	7 58 65	17 46 63	21 32 70	25 40 66	27 37 71	50 59 72
				53 61 73								
82	10	10		1 2 42	3 72 75	4 25 29	5 38 43	6 22 28	8 26 34	9 36 55	11 35 58	12 32 62
				14 31 45								
83	10	10		1 2 4	5 13 18	6 31 58	7 26 45	10 43 53	11 39 55	12 24 48	14 23 37	15 32 49
				20 41 61								
84	10	10		12 24 48	25 42 59	1 22 63	2 29 57	3 10 13	4 9 79	6 38 44	8 31 39	11 26 37
				16 35 51								
85	10	10		1 2 4	5 65 70	6 24 30	7 33 40	8 29 37	9 28 47	10 23 72	11 42 53	12 34 46
				14 41 58								
86	10	10		1 2 44	3 50 53	4 12 78	5 21 70	7 22 29	9 19 28	11 24 35	14 40 54	18 41 59
				25 31 56								
87	10	10		1 2 4	5 54 59	6 18 24	7 41 48	8 35 60	9 19 77	11 26 37	13 30 70	16 36 67
				21 43 64								
88	11	11	$\surd$	11 22 44	1 2 4	5 10 20	7 14 28	9 18 36	13 38 51	6 35 41	17 43 62	8 31 39
				12 42 54	16 32 56							
89	11	11	$\surd$	1 64 65	9 76 85	14 46 60	18 23 84	31 50 81	33 40 73	34 44 78	35 47 77	36 38 74
				41 62 68	52 69 72							
90	11	11		1 2 46	7 54 61	12 28 40	15 48 63	18 38 70	21 34 55	23 26 49	25 31 84	51 68 73
				60 71 79	66 76 80							
91	11	11		13 39 78	18 27 82	23 61 84	28 48 76	32 51 72	33 34 67	35 41 85	37 62 66	46 60 77
				69 79 81	80 83 88							
92	11	11		1 45 46	7 38 61	16 59 75	20 42 70	30 56 86	32 35 67	37 51 78	39 58 73	52 63 81
				79 83 88	80 82 90							
93	11	11		1 2 4	8 45 53	12 70 82	19 36 55	28 44 72	29 34 63	31 41 83	33 46 79	43 50 68
				54 69 78	61 67 87							
94	11	11		1 46 47	7 24 31	11 43 54	18 20 38	34 57 91	35 44 79	36 52 78	55 65 84	61 67 88
				64 72 86	12 25 53							
95	11	11		1 2 48	8 34 42	12 45 62	18 57 75	21 43 64	23 28 51	24 30 54	29 32 92	55 68 82
				70 79 86	76 80 91							
96	12	11		1 2 49	5 10 20	7 14 28	11 22 44	4 27 50	13 39 70	17 42 71	18 37 55	8 43 51
				3 6 12	16 32 72							
97	12	12	$\surd$	1 2 95	8 25 33	12 30 79	21 50 71	27 42 82	32 75 86	34 44 78	35 58 74	45 59 83
				51 60 88	56 61 92	77 84 90						
98	12	12		14 56 84	6 75 81	15 37 76	18 48 68	21 31 88	25 60 63	32 34 96	33 44 87	45 46 97
				58 74 82	72 79 91	85 89 94						
99	12	12		1 2 97	8 15 92	16 33 49	19 39 59	21 65 86	26 37 63	27 41 68	28 52 75	29 38 67
				43 48 94	54 64 89	57 69 87						
100	12	12		1 49 50	6 69 75	14 84 98	21 43 78	24 33 91	34 61 73	36 40 96	38 53 85	41 46 87
				55 74 81	72 82 90	77 80 97						
101	12	12		1 2 4	7 82 89	11 57 68	17 35 83	24 52 76	41 54 88	43 58 72	51 61 91	56 65 92
				59 64 96	62 70 93	63 79 85						
102	12	12		30 60 81	15 52 65	25 49 78	34 41 75	43 44 101	56 76 82	62 74 90	63 80 85	67 69 100
				83 91 94	84 88 98	9 32 64						
103	12	12		1 2 4	7 64 71	11 42 72	19 69 88	27 47 74	51 63 91	54 60 97	55 68 90	62 80 85
				67 77 93	70 78 95	14 38 59						
104	13	13	$\surd$	1 2 53	6 31 79	14 21 97	23 38 61	32 36 100	34 67 101	39 47 96	41 60 82	46 62 88
				54 74 84	59 77 86	69 80 93	75 87 92					
105	13	12		15 30 60	1 2 4	8 16 32	11 22 44	13 26 52	17 34 68	19 38 76	18 41 64	12 43 55
				9 49 58	6 42 69	7 14 28						
106	13	13		1 2 54	6 28 34	9 90 99	27 69 96	30 66 70	33 50 83	35 43 98	38 49 95	41 67 80
				47 61 92	62 74 94	77 82 101	85 88 103					
107	13	13		1 2 4	7 42 72	14 69 83	17 46 78	21 68 89	23 59 71	25 33 58	26 41 92	28 34 62
				37 50 87	43 53 97	67 76 98	80 91 96					
108	13	13		1 2 55	6 25 31	9 72 81	12 32 44	18 41 59	35 50 93	37 40 74	46 51 97	47 61 75
				48 70 86	52 56 82	79 87 100	84 91 101					
109	13	13		1 2 4	7 33 83	10 75 85	13 30 43	23 51 74	25 46 88	27 32 104	29 41 97	31 47 62
				36 54 90	38 49 60	39 48 100	89 95 103					
110	13	13		4 8 59	13 30 93	21 50 81	28 34 62	33 72 105	46 56 100	49 67 92	65 84 91	73 74 109
				78 87 101	83 95 98	86 88 108	11 52 68					

Table 3: Examples of best or optimal (v,4,2,1) OOC

v	s_O	s	p	codeword-sets							
111	13	13		1 17 95	14 44 58	22 49 84	28 66 73	41 46 106	48 50 98	52 71 92	54 64 101
				55 80 86	69 77 103	82 91 102	93 96 108	4 36 72			
112	14	14		48 64 80	5 81 86	12 66 78	15 37 52	17 42 87	24 43 93	27 57 84	33 53 92
				38 51 89	41 44 109	63 77 98	72 83 101	102 106 108	103 104 111		
113	14	14	√	1 2 58	7 17 103	21 60 74	24 54 83	26 72 98	33 36 69	34 79 102	35 63 85
				37 42 108	38 51 100	43 52 61	65 73 105	82 88 107	93 97 109		
114	14	13		1 2 58	5 10 20	7 14 28	11 22 44	13 26 52	17 34 68	9 41 73	8 16 69
				3 40 43	25 54 85	12 47 59	19 42 91	18 36 66			
115	14	14		1 2 4	10 27 98	11 89 100	13 18 110	14 77 91	16 35 96	22 68 90	28 34 109
				29 62 82	30 51 94	31 39 70	32 41 106	36 48 103	42 49 108		
116	14	14		1 2 59	5 65 70	7 31 92	8 37 45	9 25 100	10 52 74	15 34 97	17 20 113
				18 48 66	21 33 54	26 69 73	28 39 67	32 38 110	76 89 103		
117	14	14	√	1 2 4	6 95 101	8 44 81	9 49 58	10 35 92	15 71 86	20 27 110	21 32 53
				23 52 75	24 50 91	38 55 100	39 51 105	70 84 103	5 18 48		
118	14	14		1 2 60	5 51 72	12 16 28	14 40 54	24 30 112	27 65 80	32 63 95	33 41 74
				34 37 71	35 42 111	45 56 101	48 57 105	50 68 93	79 89 108		
119	15	15	√	17 34 85	9 83 92	14 73 87	22 35 57	25 40 104	30 61 91	33 38 71	37 56 100
				41 52 93	42 54 107	43 47 90	49 50 99	80 96 103	95 98 116	109 111 117	
120	15	14		1 2 61	7 14 28	11 22 44	13 26 52	4 31 62	17 51 86	19 57 82	9 46 55
				6 47 79	20 43 97	5 29 96	3 15 18	8 16 72	10 40 50		
121	15	15	√	1 2 62	9 67 76	14 55 80	21 65 77	31 37 115	34 38 72	36 69 105	40 51 91
				46 74 93	50 82 89	64 79 106	68 92 97	73 86 108	95 103 113	98 101 118	
122	15	15		1 60 61	18 49 67	30 76 99	35 83 118	40 52 110	44 50 116	57 71 108	68 77 113
				75 95 102	79 90 111	80 96 106	81 100 103	88 93 117	89 97 114	94 107 109	
123	15	14		1 2 4	5 10 20	7 14 28	8 16 32	11 22 44	13 26 52	17 34 68	19 38 76
				23 46 92	12 49 61	9 50 59	27 56 83	6 36 42	18 53 98		
124	15	15		1 2 63	8 29 37	18 24 118	25 70 79	40 56 96	53 64 113	58 85 97	59 74 109
				67 80 111	69 73 120	78 83 119	81 101 104	82 92 114	91 98 117	14 48 86	
125	15	15		1 2 63	9 42 51	30 68 98	35 52 108	48 70 118	53 67 111	65 78 112	66 76 115
				71 82 114	79 99 105	80 88 117	84 89 120	85 97 113	101 104 122	102 106 121	
126	16	16	√	18 36 72	1 2 64	5 10 20	11 22 44	13 26 52	17 34 68	19 38 76	3 43 83
				8 61 73	29 60 95	7 37 96	23 47 102	9 25 110	4 49 81	6 27 105	14 28 56
127	15	15		1 2 65	11 15 26	16 66 82	20 71 76	31 39 70	37 84 121	58 79 106	59 68 93
				67 81 113	74 92 109	75 87 115	83 86 124	91 98 120	94 104 117	19 49 73	
128	16	16	√	10 74 84	9 101 110	15 56 71	19 66 85	30 35 65	33 50 111	39 55 94	45 58 103
				48 86 90	51 52 127	59 88 99	60 91 97	79 102 105	82 104 106	96 108 116	100 107 114
129	16	15		1 2 4	5 10 20	7 14 28	8 16 32	11 22 44	13 26 52	17 34 68	19 38 76
				23 46 92	25 50 100	12 47 82	27 58 98	9 64 73	18 63 81	6 36 42	
130	16	16		7 72 79	9 71 80	21 38 113	25 70 85	29 49 78	32 54 86	35 41 124	46 56 120
				48 88 90	61 77 114	63 91 102	87 106 111	94 97 127	96 104 122	99 103 126	116 117 129
131	16	16		1 2 4	10 68 78	16 22 38	28 59 100	32 37 126	35 49 84	44 61 105	50 79 102
				51 62 113	54 88 97	57 76 112	64 85 110	65 92 104	71 86 101	75 83 123	98 111 118
132	16	16		1 67 68	10 61 122	16 73 89	21 33 54	28 63 91	31 84 115	45 49 94	52 74 110
				55 70 117	72 90 114	76 82 126	86 93 125	92 95 129	102 113 121	103 105 130	109 118 123
133	16	16		19 57 114	11 63 81	17 64 86	26 72 87	30 59 89	33 75 91	36 73 109	48 79 127
				49 50 99	51 53 131	56 88 101	65 93 105	71 92 112	90 98 125	110 119 124	123 126 130
134	16	16		1 2 68	4 95 99	7 54 61	11 34 111	12 81 93	13 116 129	22 60 96	28 79 107
				37 52 89	46 72 118	50 64 114	71 101 104	75 92 117	76 85 125	78 102 110	86 105 115
135	16	16		1 2 68	6 13 19	11 53 93	23 79 112	25 60 110	28 58 105	39 65 109	45 59 121
				52 86 101	57 94 98	63 81 99	64 74 125	80 92 123	97 106 126	103 111 119	108 113 130
136	17	17	√	1 67 68	2 63 65	4 20 120	9 56 89	11 32 43	14 78 92	23 31 54	25 59 84
				26 29 55	35 62 97	38 48 86	45 85 130	46 53 99	57 87 106	76 94 118	95 108 123
				114 119 131							
137	17	17	√	2 43 96	5 44 98	13 55 68	18 34 52	21 110 131	24 49 112	35 64 99	46 47 92
				51 59 129	57 77 97	58 75 133	61 71 127	65 87 115	100 111 126	101 104 134	105 114 128
				106 118 125							
138	17	17		1 35 104	2 70 136	10 61 87	11 74 85	12 60 90	13 118 131	17 62 79	23 65 96
				28 95 123	29 56 111	38 84 122	41 91 132	44 80 102	49 52 101	81 99 120	98 106 130
				119 124 133							
139	17	17		1 2 4	10 15 134	11 72 78	17 46 110	21 109 130	22 77 84	24 74 98	28 40 68
				35 66 108	38 57 76	39 92 131	43 69 113	48 80 107	54 90 103	56 81 114	79 102 116
				87 105 121							
140	17	17		20 40 80	2 59 83	4 11 133	9 32 41	12 61 73	17 52 69	19 106 125	22 76 86
				42 85 127	46 90 96	47 74 113	48 84 132	62 65 137	63 89 114	82 103 119	102 107 135
				109 110 139							

Table 4: Examples of best or optimal (v,4,2,1) OOC

v	s_O	s	p	codeword-sets							
141	17	17		1 2 4	6 103 109	8 43 106	10 52 99	11 34 118	16 65 81	17 31 48	18 72 87
				21 91 112	24 51 75	25 104 129	41 80 102	46 59 105	53 83 111	55 64 132	56 96 101
				108 115 134							
142	17	17		1 2 72	10 76 109	13 82 95	24 98 122	26 79 105	31 61 92	38 86 124	49 78 113
				54 77 131	57 96 103	59 80 121	75 100 117	84 87 139	97 106 133	102 114 130	108 123 127
				120 128 134							
143	17	17		1 3 141	6 33 116	9 13 139	14 103 117	16 51 67	18 113 131	24 63 87	28 74 97
				34 68 75	42 78 107	49 57 106	50 61 132	55 77 99	62 81 100	64 79 111	73 98 118
				91 112 122							
144	18	18	$\surd$	1 2 73	5 46 51	6 121 127	7 125 132	11 111 122	13 28 129	16 63 79	30 38 68
				31 83 92	34 48 130	35 45 134	36 39 75	49 74 123	50 54 140	53 85 112	56 80 120
				66 84 126	67 87 124						
145	18	18	$\surd$	1 3 4	9 73 82	17 75 87	22 49 118	30 36 139	35 50 130	43 71 114	46 83 108
				48 59 107	53 94 104	56 61 140	67 90 122	77 101 121	85 111 119	88 106 127	91 98 138
				93 112 126	100 116 129						
146	18	18		1 2 74	8 38 46	9 110 119	10 113 123	11 76 87	15 68 83	20 37 57	22 44 84
				24 71 99	25 105 130	34 52 86	48 67 115	56 95 107	64 69 133	85 91 140	88 92 142
				111 125 132	114 117 143						
147	18	18		21 42 84	6 13 140	8 88 96	10 66 76	11 131 142	17 102 119	18 107 125	29 64 93
				36 50 86	37 46 92	43 74 117	44 69 113	48 87 135	49 82 114	52 90 109	53 68 121
				77 100 124	143 144 145						
148	18	18		1 74 147	4 103 107	10 32 126	14 21 35	15 67 96	16 92 108	17 97 131	23 53 118
				36 64 100	39 58 129	43 46 89	47 85 110	60 69 139	61 98 111	66 86 128	91 117 122
				93 104 137	124 130 136						
149	18	18		1 13 137	11 59 101	14 44 119	18 40 58	21 50 71	23 110 133	32 35 146	34 49 134
				41 72 113	46 93 102	51 57 143	52 60 141	69 76 142	70 96 123	82 106 125	84 94 139
				88 116 121	95 112 132						
150	18	18		1 74 75	4 94 98	7 46 53	9 28 37	16 45 61	17 66 101	21 83 88	24 68 92
				32 80 102	38 110 148	50 93 107	55 63 142	59 85 144	69 103 116	86 109 127	99 119 130
				108 123 135	114 117 147						
151	18	18		1 8 144	6 9 148	11 41 52	16 122 138	19 115 134	22 56 78	25 75 101	31 70 112
				32 65 118	35 92 127	37 64 124	38 48 141	63 67 130	69 74 146	71 89 133	83 97 137
				85 108 128	100 102 149						
152	19	19	$\surd$	2 76 150	5 13 144	6 57 101	11 47 116	17 75 94	18 45 63	20 70 102	22 68 106
				28 31 149	30 42 140	33 93 126	49 86 115	56 80 136	65 88 129	71 85 138	73 112 113
				90 97 145	91 100 143	117 127 142					
153	19	19	$\surd$	1 2 77	9 26 35	15 22 146	21 61 82	23 36 140	24 69 108	31 50 81	41 68 126
				43 94 137	46 83 116	47 57 143	53 97 109	54 74 128	55 73 135	63 105 111	65 93 125
				87 101 139	89 119 123	142 145 150					
154	19	19		22 44 88	2 33 123	3 72 75	5 55 60	11 28 137	15 56 113	19 86 87	20 81 93
				43 51 146	46 80 120	47 84 131	53 71 136	58 96 106	62 76 138	91 118 127	100 124 130
				102 115 128	105 112 147	125 129 150					
155	19	19		1 2 4	6 36 42	7 76 86	11 52 114	12 26 141	13 122 135	17 60 77	22 124 146
				24 80 99	25 92 117	28 51 132	34 100 134	37 110 147	47 65 137	49 54 150	57 84 128
				58 87 116	70 85 120	91 107 123					
156	19	19		1 2 79	7 102 109	10 74 92	11 73 94	13 110 123	15 81 96	17 39 56	19 132 151
				25 28 53	31 40 71	32 55 87	35 76 111	48 99 105	49 65 114	50 88 118	52 86 122
				67 93 130	84 98 142	136 144 148					
157	19	19		1 2 4	6 28 135	7 67 97	8 59 106	9 27 36	11 54 65	12 116 128	14 24 38
				31 57 131	34 86 120	42 91 108	45 64 109	46 78 124	55 70 142	56 96 117	62 75 137
				63 88 132	68 73 141	76 99 134					
158	19	19		1 2 80	4 127 131	7 71 94	8 63 103	9 33 134	10 48 58	12 41 53	13 26 132
				14 75 89	16 76 98	20 88 108	32 43 147	37 73 122	56 102 130	62 107 113	65 99 124
				67 72 139	81 111 128	137 140 155					
159	19	19		1 2 157	5 89 94	8 25 142	9 111 120	11 22 96	15 46 61	20 115 135	28 86 114
				35 69 125	36 54 141	41 81 119	43 49 92	50 77 132	68 97 130	76 88 147	79 93 145
				106 129 136	108 127 140	10 26 47					
160	20	20	$\surd$	1 2 81	4 101 105	6 16 22	9 95 104	11 121 132	14 32 46	17 34 126	25 62 123
				29 60 129	30 77 113	38 43 155	40 94 134	41 76 125	57 102 115	64 72 136	70 89 141
				75 78 153	87 107 140	93 116 137	118 133 145				
161	20	20		23 46 92	3 57 107	4 99 103	8 34 135	10 39 49	13 76 89	19 140 159	24 31 154
				25 53 78	30 45 75	36 71 126	37 51 88	41 59 143	65 87 139	67 94 114	81 93 149
				82 91 152	97 113 129	117 118 160	144 150 155				

Table 5: Examples of best or optimal (v,4,2,1) OOC

v	s_O	s	p	codeword-sets							
162	20	20		1 2 82	4 108 112	6 18 24	7 128 135	10 76 96	14 75 101	15 74 89	16 114 146
				19 68 113	23 56 79	29 40 151	30 92 100	31 126 157	38 55 93	43 85 120	44 65 109
				57 60 159	71 110 123	90 115 137	116 125 153				
163	20	20		1 2 4	8 76 95	11 43 54	13 27 149	16 86 102	22 31 53	25 49 74	28 34 157
				29 58 125	33 83 113	39 65 104	47 84 126	48 71 140	63 107 119	64 82 146	66 112 117
				73 108 128	85 106 142	88 103 148	91 101 153				
164	20	20		1 2 83	4 89 93	7 42 129	8 30 38	9 68 105	11 32 43	12 24 140	16 33 49
				19 34 53	26 70 120	27 73 118	28 67 95	29 80 109	52 58 110	57 104 161	62 103 123
				64 78 150	74 99 139	88 98 154	133 146 151				
165	20	20		1 3 164	5 69 101	10 43 53	18 107 125	20 36 149	24 87 102	28 84 109	30 65 95
				31 48 79	34 108 142	38 60 143	45 106 151	49 75 124	50 97 118	67 94 138	72 80 157
				92 103 154	99 111 153	110 119 156	146 152 159				
166	20	20		1 2 84	4 26 144	6 67 105	9 103 112	16 28 44	17 35 148	20 109 129	23 47 70
				27 91 102	31 121 152	33 120 153	36 68 104	41 56 151	50 93 123	52 92 126	55 60 115
				58 66 124	59 80 145	78 81 159	117 127 156				
167	20	20		1 2 4	5 59 118	7 107 114	9 48 128	12 155 161	19 36 55	22 101 145	25 45 147
				27 62 97	28 92 120	29 124 153	33 63 137	40 90 130	41 56 152	51 83 135	57 91 133
				68 106 129	69 82 151	73 81 159	136 146 157				
168	21	20		24 48 96	1 2 85	5 10 20	11 22 44	13 26 52	17 34 68	19 38 76	4 45 86
				23 69 122	25 75 118	27 74 101	3 58 61	7 73 102	37 77 114	16 65 81	9 80 89
				28 59 137	8 70 106	6 18 156	14 35 147				
169	21	21	$\surd$	3 60 112	7 74 102	9 89 160	11 27 38	14 65 118	19 31 157	22 56 78	30 75 105
				35 76 111	37 77 114	39 68 107	46 90 136	47 106 110	48 121 145	49 50 99	54 69 154
				61 82 148	81 83 164	127 144 152	133 146 156	137 143 163			
170	21	21		4 81 85	7 140 147	10 18 162	11 69 112	12 83 99	15 55 130	19 51 70	20 22 42
				21 124 145	27 86 113	31 76 125	33 39 72	34 82 116	35 38 167	60 74 134	68 97 165
				90 118 142	91 104 157	92 109 153	105 114 161	106 107 169			
171	21	21		2 21 23	6 24 153	8 33 146	9 80 100	10 89 99	11 73 109	14 65 120	15 27 159
				28 69 130	29 86 115	34 87 121	37 59 96	38 101 139	43 88 131	60 107 124	67 93 119
				68 81 158	74 105 140	95 125 141	117 122 166	164 167 168			
172	21	21		2 88 90	3 92 95	5 98 103	7 20 27	9 33 42	10 147 157	12 56 68	21 87 106
				22 48 70	36 99 135	39 57 96	41 81 122	45 75 142	49 107 156	53 64 117	54 71 125
				60 61 171	94 126 140	100 134 138	113 121 164	120 143 149			
173	21	21		1 7 8	9 12 21	13 43 56	14 97 111	15 121 136	18 78 113	20 31 162	23 73 146
				41 65 149	46 75 144	47 105 115	48 80 128	49 54 103	51 110 114	53 87 140	66 91 157
				71 109 135	77 79 171	99 118 154	112 134 151	28 72 116			
174	21	21		1 2 88	3 79 98	5 83 96	8 51 131	9 59 124	12 130 142	15 119 134	18 63 129
				20 113 133	24 31 167	26 36 164	27 74 101	28 62 90	33 102 135	42 106 110	49 97 126
				53 67 120	89 114 149	92 122 144	116 137 153	151 157 168			
175	22	22	$\surd$	25 50 100	3 37 40	5 82 87	6 58 64	7 83 99	8 81 102	14 71 85	17 130 147
				18 137 155	23 47 70	30 79 109	31 134 165	32 78 110	36 91 120	42 122 164	44 59 160
				60 114 174	63 106 132	89 108 156	107 140 142	124 136 163	153 162 166		
176	22	22	$\surd$	1 2 89	4 39 141	5 63 118	6 40 46	10 83 103	11 86 101	12 21 167	16 77 115
				17 42 151	18 65 129	24 43 157	27 98 105	31 45 76	32 91 123	44 80 140	48 51 173
				56 82 150	66 79 163	69 106 139	95 124 147	104 126 154	119 127 168		
177	22	21		1 2 4	5 10 20	7 14 28	8 16 32	11 22 44	13 26 52	17 34 68	19 38 76
				23 46 92	25 50 100	29 58 116	31 62 124	35 70 140	18 65 112	12 67 79	9 80 89
				30 73 103	40 81 121	27 86 113	36 78 114	6 60 66			
178	22	22		1 2 90	4 37 145	5 62 121	7 135 142	8 64 122	9 158 167	12 18 24	17 84 111
				23 78 101	25 73 98	32 127 159	34 69 103	53 93 138	54 96 150	63 110 131	65 106 137
				71 97 168	86 99 165	87 102 163	104 134 148	118 140 156	126 129 175		
179	22	22		1 2 4	5 73 111	7 143 150	8 93 101	9 44 144	10 42 52	11 142 153	12 84 107
				13 132 145	15 77 117	18 64 133	27 98 125	28 49 158	33 122 155	53 112 165	55 80 154
				58 75 162	83 103 159	88 118 149	97 113 163	116 138 157	128 134 173		
180	22	22		1 2 91	4 38 146	5 60 125	7 128 135	9 12 21	13 71 84	14 117 131	19 94 105
				23 127 150	24 74 130	25 61 144	32 78 110	40 73 113	43 80 123	51 69 162	62 101 163
				66 88 154	81 116 145	82 126 136	85 112 153	132 152 160	133 149 164		
181	22	22		1 2 4	5 67 124	6 89 95	8 33 156	10 82 109	15 76 120	18 29 170	20 59 79
				22 63 85	32 56 88	34 46 80	37 91 127	42 55 168	43 64 107	45 110 155	49 84 133
				77 94 164	78 106 153	81 100 131	112 121 172	115 129 167	128 151 158		

All computer results are obtained by our own C++ programs. Since programming mistakes are always possible, we partially checked the correctness of the classification software by obtaining for several small values of v all OOCs (including the isomorphic ones), and then applying to the related k-resolutions of partial 2-(v,4,2) designs a programme filtering away isomorphic resolutions. The number of nonisomorphic k-resolutions obtained this way was the same as the number of all nonisomorphic OOCs.

We also checked if our results are consistent with some previous results of other authors, namely

- It is proved in [3] that there is no optimal (v, 4, 2, 1)-OOC in each of the following cases: $v = 24$ or 33 (mod 72); $v = 42$ or 48 or 51 or 57 (mod 72) but $v \neq 0$ (mod 7).

  Classifying (v, 4, 2, 1)-OOCs with $v \leq 75$ we obtain that there are no optimal codes for all lengths following from the statement above, namely $24, 33, 48, 51, 57$, and also for four other lengths, namely $16, 18, 27$ and 32.

  Only for lengths $96, 105, 114, 120, 123, 129, 168$ and 177 (nonexistence of optimal codes proved in [3] for all these values) we could not find examples of optimal (v, 4, 2, 1)-OOCs, and so we present best OOCs in the tables.

- An optimal (v, 4, 2, 1)-OOC is necessarily perfect [3] in the following cases: $v = 7$ (mod 56); $v = 14$ (mod 56); $v = 1$ (mod 8) but $v \neq 49$ (mod 56); $v = 0$ (mod 8) but $v \neq 0$ (mod 56).

  We obtain only perfect codes for lengths $25, 40, 41, 63, 64, 65, 70, 72, 73$. The arbitrary codes we have constructed for lengths $80, 81, 88, 89, 97, 104, 113, 119, 121, 126, 128, 136, 137, 144, 145, 152, 153, 160, 169, 175$ and 176 are perfect.

- We also checked if some of the codes we construct include codes following from known constructions. For instance it is proved in [3] that there exists a perfect (8p, 4, 2, 1)-OOC for every prime $p = 1$ (mod 6). This code has one codeword-set whose differences are precisely the non-zero elements of the subgroup of order 8 of Z_{8p}. So we looked for such codes among the constructed (56, 4, 2, 1)-OOCs and found out that about 1/3 of them have such a codeword-set.

In conclusion, we believe that the classified in this work codes will be of use both directly and as ingredients in constructions of new infinite families.

References

[1] Ts. Baicheva, S. Topalova, Classification of optimal $(v, 4, 1)$ optical orthogonal codes with $v \leq 76$, Proceedings of the 12th International Workshop on Algebraic and Combinatorial Coding Theory,

Novosibirsk, Russia (2010), 26-30.

[2] I.C.M. Bird, A.D. Keedwell, Design and applications of optical orthogonal codes - a survey, *Bull. Inst. Comb. Appl.* 11, 2144 (1994).

[3] M. Buratti, K. Momihara, A. Pasotti, New results on optimal (v, 4, 2, 1) optical orthogonal codes, *Des. Codes Cryptography*, Volume 58, Number 1 (2011), 89-109.

[4] W. Chu , C.J. Colbourn, Optimal (n, 4, 2)- OOC of small order, *Discrete Math.* 279(2004), 163 – 172.

[5] F.R.K. Chung , J.A. Salehi , V.K. Wei, Optical orthogonal codes: design, analysis and applications, *IEEE Trans. Inform. Theory* 35 (1989), 595 - 604 .

[6] Ch. Colbourn, J. Dinitz editors, Handbook of Combinatorial Designs, 2nd edition (Discrete mathematics and its applications, ser. ed. K. Rosen), CRC Press, Boca Raton, FL., 2007.

[7] C.J. Colbourn , J.H. Dinitz , D.R. Stinson , Applications of combinatorial designs to communications, cryptography, and networking, in J.D. Lamb, D.A. Preece (eds.) Surveys in combinatorics, pp. 37 - 100. Cambridge University Press, London (1999).

[8] M.J. Colbourn, R.A. Mathon, On cyclic Steiner 2-designs, Ann. Discrete Math. 7 (1980), 215 - 253.

[9] T.C. Frenz, Computing Techniques for the Enumeration of Cyclic Steiner Systems, Ph.D. Thesis, Rochester Institute of Technology School of Computer Science, 1989.

[10] R. Fuji-hara, Y. Miao, Optical orthogonal codes: Their bounds and new optimal constructions, *IEEE Trans. Inform. Theory* vol. 46 (2000), 2396 – 2406.

[11] P. Kaski, P. Östergård, *Classification algorithms for codes and designs*, Springer, Berlin, 2006.

[12] K. Momihara, M. Buratti, Bounds and Constructions of Optimal (n, 4, 2, 1) Optical Orthogonal Codes, *IEEE Trans. Inform. Theory* vol. 55 (2009), 514 - 523.

[13] Muzychuk M., A solution of the isomorphism problem for circulant graphs, Proc. London Math. Soc. (3) 88 (2004), 1 - 41.

[14] D. R. Stinson , R. Wei , J. Yin , Packings, in C.J. Colbourn, J.H. Dinitz (eds.), The CRC handbook of combinatorial designs, 2nd edn., pp. 550 - 556. Chapman and Hall/CRC Press, Boca Raton, FL (2006).

[15] G.C. Yang , T.E. Fuja, Optical orthogonal codes with unequal auto- and cross-correlation constraints, *IEEE Trans. Inform. Theory* vol. 41 (1995), 96 - 106.