New lower bounds on error-correcting ternary, quaternary and quinary codes

Antti Laaksonen & Patric R. J. Östergård
Aalto University

Definitions

- An $(n,M,d)_q$ code is a q-ary code with length n, size M and minimum distance d
- Example: a $(5,3,2)_4$ code:

  {00000, 01020, 33311}

Definitions

- $A_q(n,d)$ denotes the maximum size M of an $(n,M,d)_q$ code
- Example: there is a $(5,3,2)_4$ code, so we know that $A_4(5,2) \geq 3$

  {00000, 01020, 33311}

Goal

- Find new codes that yield lower bounds for $A_q(n,d)$
- We focus on cases where $3 \leq q \leq 5$

Example: case q=3

	3	4	5	6	7	8	9	10	11	12	13	14	15
4	9												
5	18	6											
6	38	18	4										
7ub	111	33	10	3									
7lb	99	=	=	=									
8ub	333	99	27	9	3								
8lb	252	=	=	=	=								
9ub	937	297	81	27	6	3							
9lb	729	243	=	=	=	=							
10ub	2808	891	243	81	14	6	3						
10lb	2187	729	=	=	=	=	=						
11ub	7029	2561	729	243	36	12	4	3					
11lb	6561	1458	=	=	=	=	=	=					
12ub	19683	6839	1557	729	108	36	9	4	3				
12lb	=	4374	729	=	60	=	=	=	=				
13ub	59049	19270	4078	1449	324	95	27	6	3	3			
13lb	=	8559	2187	729	162	54	=	=	=	=			
14ub	153527	54774	10624	3660	805	237	62	13	6	3	3		
14lb	118098	24786	6561	2187	243	108	36	=	=	=	=		
15ub	434815	149585	29213	9904	2204	685	165	39	10	6	3	3	
15lb	354294	72171	6561	2187	729	243	81	24	=	=	=	=	
16ub	1240029	424001	77217	27356	6235	1923	451	114	29	9	4	3	3
16lb	1062882	216513	19683	6561	729	297	243	54	18	=	=	=	=

https://www.win.tue.nl/~aeb/codes/ternary-1.html

General ideas

- Focus on codes that have symmetries
- Use a special representation for codewords
- Use computer search to automatically find codes

Representation

- A *q*-ary codeword of length *n* is an *n*-element subset of {1,2,...,*nq*}
- Example: codeword 0120 = {1,4,6,11}
-

index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
symbol	0	0	0	0	1	1	1	1	2	2	2	2
position	1	2	3	4	1	2	3	4	1	2	3	4
added?	X			X		X					X	

Permutations

- Idea: modify codewords using permutations
- In our new representation, we can both permute (1) coordinates and (2) coordinate values
- Examples:
  - (1) 0120 → 1020
  - (2) 0120 → 2120

Permutations

- How to use permutations?
- Example: permutation (1 5 9)
  initial codeword 0120 = {1,4,6,11}

index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
symbol	0	0	0	0	1	1	1	1	2	2	2	2
position	1	2	3	4	1	2	3	4	1	2	3	4
added?	X			X		X					X	

Permutations

- How to use permutations?
- Example: permutation (1 5 9)
  new codeword 1120 = {4,5,6,11}

index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
symbol	0	0	0	0	1	1	1	1	2	2	2	2
position	1	2	3	4	1	2	3	4	1	2	3	4
added?				X	X	X					X	

Permutations

- Not all permutations are valid
- Example: permutation (1 2)
  initial codeword 0120 = {1,4,6,11}

index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
symbol	0	0	0	0	1	1	1	1	2	2	2	2
position	1	2	3	4	1	2	3	4	1	2	3	4
added?	X			X		X					X	

Permutations

- Not all permutations are valid
- Example: permutation (1 2)
  new codeword ?!?! = {2,4,6,11}

index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
symbol	0	0	0	0	1	1	1	1	2	2	2	2
position	1	2	3	4	1	2	3	4	1	2	3	4
added?		X		X		X					X	

Groups

- Idea: focus on codes that are prescribed by a permutation group
- Such codes consist orbits of codewords
- Example: $G = \langle (1\ 5\ 9) \rangle$

  orbit repr.: {0000,1111,2222}

  final code: {0000,1000,2000,
  0111,1111,2111,
  0222,1222,2222}

Groups

- A group can only be used if its permutations are valid, i.e., it has a block system where each block size is *q*
- A group may have several such block systems
- Example: G = ⟨(1 5 9)⟩

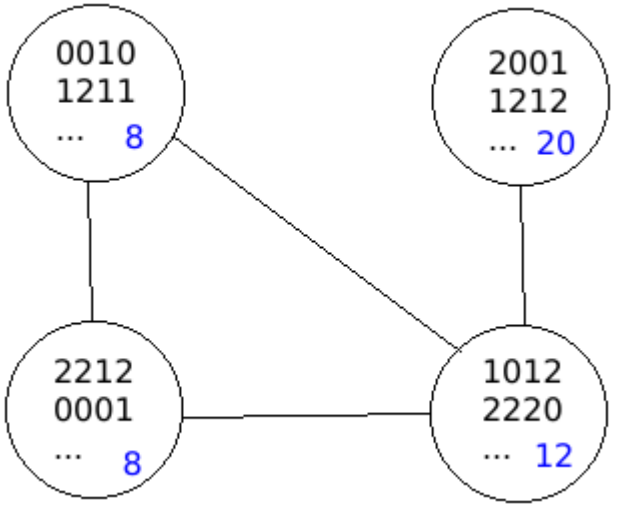
index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
symbol	0	0	0	0	1	1	1	1	2	2	2	2
position	1	2	3	4	1	2	3	4	1	2	3	4

Computer search

- Idea: use a computer to go through a large number of potential groups
- For each group, consider its possible block systems separately
- Remaining task: how to choose the orbits that will create the code?

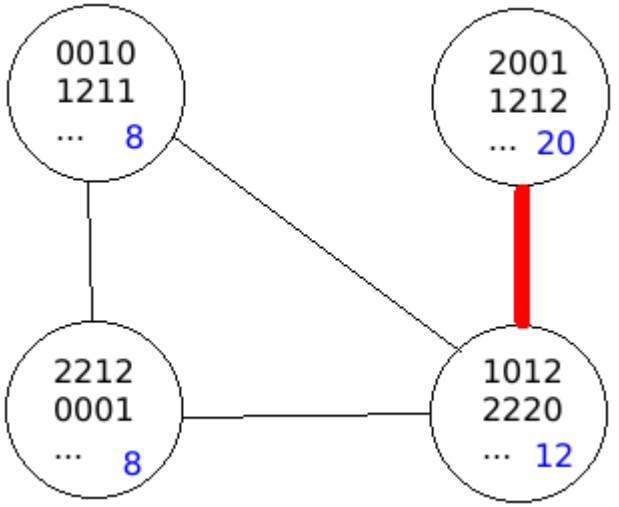
Computer search

- The orbit selection problem can be modelled as a maximum weight clique problem in a graph
- Example:



Computer search

- The orbit selection problem can be modelled as a maximum weight clique problem in a graph
- Example:



Computer search

- However: the maximum weight clique problem is NP-hard in the general case
- Still, we can try to solve it using, for example, backtracking
- Since each node in the graph is an orbit, the graph is not so large

Computer search

- Idea: focus on *transitive* permutation groups
- Reasons:
  - many existing codes have such symmetries
  - the groups have been classified up to degree 47
  - the number of such groups is not too large

Computer search

- We went through transitive permutation groups up to degree 47
- We used the *Cliquer* software to construct maximum weight cliques
- Variations: divide each codeword into two parts, fix one coordinate, or both
- The search took about one week on a cluster

Results

- **old result**

  $A_3(13,4) \geq 8559$
  $A_3(14,4) \geq 24786$
  $A_3(15,4) \geq 72171$
  $A_3(15,5) \geq 6561$
  $A_3(15,6) \geq 2187$
  $A_3(16,7) \geq 729$
  $A_3(16,8) \geq 297$
  $A_4(8,4) \geq 320$
  $A_4(8,5) \geq 70$
  $A_4(9,4) \geq 1024$

new result

$A_3(13,4) \geq 13122$
$A_3(14,4) \geq 27702$
$A_3(15,4) \geq 83106$
$A_3(15,5) \geq 7812$
$A_3(15,6) \geq 3321$
$A_3(16,7) \geq 1026$
$A_3(16,8) \geq 387$
$A_4(8,4) \geq 352$
$A_4(8,5) \geq 76$
$A_4(9,4) \geq 1152$

Results

- **old result**

  $A_4(9,6) \geq 64$
  $A_4(10,3) \geq 17408$
  $A_4(10,4) \geq 4096$
  $A_4(11,3) \geq 65536$
  $A_5(8,4) \geq 1125$
  $A_5(8,5) \geq 160$
  $A_5(9,4) \geq 3750$
  $A_5(9,5) \geq 625$
  $A_5(10,4) \geq 15625$

new result

$A_4(9,6) \geq 76$
$A_4(10,3) \geq 24576$
$A_4(10,4) \geq 4192$
$A_4(11,3) \geq 77056$
$A_5(8,4) \geq 1225$
$A_5(8,5) \geq 165$
$A_5(9,4) \geq 4375$
$A_5(9,5) \geq 725$
$A_5(10,4) \geq 17500$

One detail

- Old trick: after creating a code, try to add more codewords
- Initial result: $A_3(15,5) \geq 7452$
  Improved result: $A_3(15,5) \geq 7812$
  (360 added codewords)
- We don't know if we could add more than 360 extra codewords

Is this all?

- We could process *almost* all transitive groups up to degree 47
- However sometimes we couldn't, because:
  - the graph would have been too large (we had a limit of 5000 nodes), or
  - the clique search still took too much time even if the graph was not so large

Is this all?

- So it is possible that we missed some codes that would yield new records
- Challenge: how to process the remaining groups?


That's all Folks!