Sixth International Workshop on Optimal Codes and Related Topics
June 16-22, 2009, Varna, Bulgaria *pp. 179-183*

Methods for equidistant code search in computer package QPlus

Todor Todorov, Galina Bogdanova
todor@moi.math.bas.bg, galina@moi.math.bas.bg
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, P.O. Box 323, 5000 Veliko Tarnovo, Bulgaria.

Abstract. New tools in computer package for coding theory research and studying QPlus are presented. QPlus includes a DLL library package that implements coding theory algorithms. We consider some methods for searching bounds on the size of q-ary equidistant codes by computer methods. Some examples for optimal equidistant codes and constant-weight equidistant codes that have been constructed by computer methods developed in QPlus are described.

1 Introduction

The main subject of the research in this work are computer methods for searching of equidistant codes.

Let us introduce some basic notations which we need to describe the results. Let Q be an alphabet of $q \geq 2$ elements. We consider the set Q^n, consisting of ordered n-tuples of elements of Q. The *Hamming distance* between two n-tuples of Q^n is defined as the number of coordinates, in which they differ. We call any subset of Q^n a *q-ary code of length n* or simply a *code* over the alphabet Q. The elements of a code are called codewords. An important parameter of each code is its minimum distance - the smallest possible Hamming distance between two different codewords. An *$(n, M, d)_q$-code* is a q-ary code of length n containing M codewords, and of minimum distance d. The alphabet Q can consist of the elements of the set $Z_q = \{0, 1, \ldots, q-1\}$. The *Hamming weight* of an n-tuple of Z_q^n is defined as the number of its nonzero coordinates. We call *a constant-weight code* the (n, M, d)-code in which every codeword has a Hamming weight w.

An $(n, M, d)_q$ *equidistant code* is a set of M codewords of length n over the alphabet $\{0, 1, ..., q-1\}$, such that any two codewords differ in d positions.

One of the fundamental problems in Coding Theory is to find $A_q(n, d)$ - the largest value of M for which there exists an $(n, M, d)_q$-code.

Codes with parameters $(n, A_q(n, d), d)_q$ are called *optimal*. In a similar manner we consider the function $B_q(n, d, w)$ for constant-weight codes and the function $B_q(n, d)$ for equidistant codes.

Let B be an ordered base $b_1, b_2, ..., b_n$ over Z_q^n and let $x = \lambda_1 b_1 + \lambda_2 b_2 + ... + \lambda_n b_n$ and $y = \mu_1 b_1 + \mu_2 b_2 + ... + \mu_n b_n$ be vectors from Z_q^n.

We say that x precedes y in lexicographical order if $(\lambda_1, \lambda_2, .., \lambda_n)$ precedes $(\mu_1, \mu_2, .., \mu_n)$ in lexicographical order, i.e. $\lambda_1 \leq \mu_1, \lambda_2 \leq \mu_2, .., \lambda_n \leq \mu_n$.

Lexicographic codes of length n and Hamming distance d are obtained by considering all q-ary vectors with weight w in lexicographic order, and adding them to the code if they are at a distance exactly d from the words that have been added earlier.

The aim of the computer systems related to the research in Coding Theory is to facilitate some routine work in the q-ary codes research. The idea of using a computer for this purpose is not a new one. Among the computer systems in this area is the program package for research on linear codes GUAVA [10]. The program system GFQ for calculations over finite fields with its modifications - an object-oriented library $GF2$ for calculations over finite fields with characteristics 2 were developed at the Applied Mathematics and Informatics Laboratory [1]. In [9] the system LinCoR for the study of binary linear codes is presented. The system QLC for studying q-ary linear codes [8] represents a development of BLC [6] [7]. The QCC is program for finding q-ary constant-weight codes from other codes [11]. In [2] system for linear codes studying Q-Extension is presented.

In Section 3 implementation of methods for equidistant code search program in computer package QPlus is presented. Some example search results obtained by QPlus are described.

2 Equidistant codes searching

Algorithm 1 Equidistant code searching
Input. Number of symbols q in the alphabet and length n, distance d of the searched code.
Output. Equidistant codes with the searched parameters and maximum number of codewords M if any exists.
Procedure:

1) Initialization. Read all the input data. Fix first two codewords of the searched code with
$a = 0, 0, ...0$ and $b = \underbrace{0, 0, ..., 0}_{n-d}, \underbrace{1, 1, ..., 1}_{d}$

Codeword representation: Let we have a codeword $x = x_1, x_2, ...x_n$. We use the following internal representation of the codeword x in our algorithm: $x_d = \{(i, x_i) : x_i \neq 0, i \in \{1...n\}\}$

2) Generate vector space of all possible codewords. These are all the codewords in lexicographic order between that are on distance d from the two fixed codewords a and b in the code.

3) Perform a backtrack search - add a codeword from generated vector space to the code if the distance between all the codewords in the code remains equal to d and the newly added codeword doesn't break lexicographic order of the columns in the code.

Two columns $b = \{b_1, b_2, ..., b_M\}$ and $c = \{c_1, c_2, ..., c_M\}$(b precedes c in the code matrix) of a code have good lexicographic order of columns if $b_i = c_i, i = 1...k, k \leq M$ and $b_{k+1} \leq c_{k+1}$

4)If we reach the end of the space we check if the size of the newly founded code is bigger than the best code that we have up to this moment. If yes then the newly founded code becomes best code. Finally we are doing a step back and change the codewords on the previous levels.

Algorithm 2 Lexicographic constant weight equidistant code searching

Input. Number of symbols q in the alphabet and length n, distance d, weight w of the searched code.

Output. Equidistant codes with the searched parameters and maximum number of codewords M if any exists.

Procedure:

1) Initialization. Read all the input data. Fix codewords that are included in the seed.

Codeword representation: Let we have a codeword $x = x_1, x_2, ...x_n$. We use the following internal representation of the codeword x in our algorithm: $x_d = \{(i, x_i) : x_i \neq 0, i \in \{1...n\}\}$

2) Generate vector space of all possible codewords. First we remove all the codewords that don't have weight w. Then we take all the codewords in lexicographic order that are on distance d from the seed.

3) Perform a greedy search - add a codeword from generated vector space to the code if the distance between all the codewords in the code remains equal to d and the newly added codeword doesn't break lexicographic order of the columns in the code.

Two columns $b = \{b_1, b_2, ..., b_M\}$ and $c = \{c_1, c_2, ..., c_M\}$(b precedes c in the code matrix) of a code have good lexicographic order of columns if $b_i = c_i, i = 1...k, k \leq M$ and $b_{k+1} \leq c_{k+1}$

4) No backtracking. If we reach the end of the space we output the code founded.

The algorithm has the following options:

1) Automatic search with each of the possible seeds with given size and found the best code.

2) Search with cycle shift from the initial space which appears to produce better codes then the standard space order.

3 Some results obtained by QPlus

The QPlus system offers computations over $Z_q = \{0, 1, \ldots, q-1\} (q < 256)$ and includes modular arithmetic, elementary number theory, vectors and matrices arithmetic, codes. The application can be successfully used for research and educational purposes. We use Delphi's ActiveForm technology and QPlus DLL library package to create a secure WEB-based system for coding theory studying. The system is an improved Web-based version of QPlus [3] and QCC [11]. We also create two new modules in the system: code search(equidistant, constant-weight equidistant and lexicographic equidistant codes) and code equivalence. QPlus searching tools are used to find many of the codes in research papers devoted to equidistant codes [5], [4]. Next are presented some examples of QPlus searching methods.

Example 1
Backtrack search construction

We can construct equidistant code with parameters $(4, 9, 3)_4$ by fixing codewords $(0, 0, 0, 0)$ and $(0, 0, 0, 1)$ and apply Algorithm 1.

We obtain following optimal code:

{0000, 0111, 0222, 1012, 1120, 1201, 3021, 3102, 3210}.

Example 2
Construction with extension

We can construct equidistant code with parameters $(6, 25, 5)_5$ by adding starting zero to each codeword of $(5, 5, 5)_5$ code and apply Algorithm 1 using produced $(6, 5, 5)_5$ code. This modification of algorithm is also included in QPlus.

Thus we obtain following optimal code:

{000000, 011111, 022222, 033333, 044444, 101234, 112340, 123401, 134012, 140123, 202413, 213024, 224130, 230241, 241302, 303142, 314203, 320314, 331420, 342031, 404321, 410432, 421043, 432104, 443210}.

Example 3
Construction of lexicographic code

We can construct equidistant code with parameters $(6, 9, 5)_4$ by using codeword $(1, 0, 1, 1, 1, 1)$ as a seed and apply Algorithm 2.

We obtain following optimal constant-weight code:

{101111, 011222, 022133, 113303, 220312, 232201, 323021, 331330}.

We can obtain corresponding equidistant code by adding all zero codeword to the code.

References

[1] Ts. Baicheva, G. Bogdanova, S. Ilieva, S. Topalova, Object-oriented C++ library for computations in and over finite fields of characteristic 2, *Math. Educ. Math.* 23, 1994, 227-230.

[2] I. Bouyukliev, About the code equivalence, *Adv. Coding Theory Cryptol., Ser. Coding Theory Cryptol.* 3, Hackensack: World Scientific, 3, 2007, 126-151.

[3] G. Bogdanova, S. Kapralov, V. Todorov, T. Parvanov, QPlus - a computer package for coding theory research, *Math. Educ. Math.* 32, 2003, 233-238.

[4] G. Bogdanova, T. Todorov, T. Pagkou, New ternary and quaternary equidistant constant weight codes, Preprint 1/2008, IMI-BAS, 2008.

[5] G. Bogdanova, T. Todorov, V. Zinoviev, On construction of q-ary equidistant codes, *Probl. Inform. Transm.* 43, 2007, 13-36.

[6] S. Kapralov, P. Petrov, Program tool for binary linear codes researches, *Proc. Conf. TU Gabrovo*, 1992.

[7] S. Kapralov, P. Petrov, P. Christov, QLC - Program tool for q-ary linear codes research, *Math. Educ. Math.* 23, 1994, 271-227.

[8] S. Kapralov, P. Christov, G. Bogdanova, The new version of QLC – a computer program for linear codes studying, *Proc. Intern. Workshop OCRT*, Sozopol, Bulgaria, 1995, 11-14.

[9] K. Manev, LINCOR – a system for linear codes researches, *Math. Educ. Math.* 26, 1987, 500-503.

[10] J. Simonis, GUAVA, A computer algebra package for coding theory, *Proc. Fourth Intern. Workshop ACCT*, Novgorod, Rusia, 1994, 165-167.

[11] V. Todorov, QCC – constructing codes from other codes, Student's Section Spring Conf. UBM, Borovec, Apr. 3-6, 2001.