Heuristic Construction of Linear Codes with prescribed Automorphism Group

Johannes Zwanzger

University of Bayreuth

Soria Summer School
July 2008

- A linear code C over $\mathbb{F}_q$ of *blocklength* n and *dimension* k is a $k-$dimensional subspace of $\mathbb{F}_q^n$

- A linear code C over $\mathbb{F}_q$ of *blocklength* n and *dimension* k is a $k-$dimensional subspace of $\mathbb{F}_q^n$
- elements of C are called *codewords* and written as *row vectors*

- A linear code C over $\mathbb{F}_q$ of *blocklength* n and *dimension* k is a $k-$dimensional subspace of $\mathbb{F}_q^n$
- elements of C are called *codewords* and written as *row vectors*
- weight $wt(c)$ of $c \in C$: number of nonzero components in c

- A linear code C over $\mathbb{F}_q$ of *blocklength* n and *dimension* k is a $k-$dimensional subspace of $\mathbb{F}_q^n$
- elements of C are called *codewords* and written as *row vectors*
- weight $wt(c)$ of $c \in C$: number of nonzero components in c
- *Hamming distance* between c, $c' \in C$:
$dist(c, c') := wt(c - c')$

- A linear code C over $\mathbb{F}_q$ of *blocklength* n and *dimension* k is a $k-$dimensional subspace of $\mathbb{F}_q^n$
- elements of C are called *codewords* and written as *row vectors*
- weight $wt(c)$ of $c \in C$: number of nonzero components in c
- *Hamming distance* between c, $c' \in C$: $dist(c, c') := wt(c - c')$
- The *minimum distance* of C is the minimum Hamming distance between any two *different* codewords of C.

- A linear code C over $\mathbb{F}_q$ of *blocklength* n and *dimension* k is a $k-$dimensional subspace of $\mathbb{F}_q^n$
- elements of C are called *codewords* and written as *row vectors*
- weight $wt(c)$ of $c \in C$: number of nonzero components in c
- *Hamming distance* between c, $c' \in C$: $dist(c, c') := wt(c - c')$
- The *minimum distance* of C is the minimum Hamming distance between any two *different* codewords of C.
- C has minimum distance $d \Rightarrow$ up to $\lfloor \frac{d-1}{2} \rfloor$ errors can be corrected

Lemma

Existence of a linear k-dimensional code over $\mathbb{F}_q$ with blocklength n and minimum distance d

$$\Updownarrow$$

Existence of a (multi-)set P of n points in $PG(k-1, q)$ so that for every hyperplane H holds: $|H \cap P| \leq n - d$.

Corollary

The search of linear (n, k, d, q)-codes is equivalent to looking for solutions of the following diophantine (in-)equation system:

$$\begin{aligned} M_q^k x &\leq \begin{pmatrix} n-d \\ n-d \\ \vdots \\ n-d \end{pmatrix} \\ \mathbb{1}^T x &= \quad n \end{aligned}$$

where $x \in \mathbb{N}_0^m$ and M_q^k is the $m \times m$ incidence matrix between points (columns) and hyperplanes (rows) in $PG(k-1, q)$.

Corollary

The search of linear (n, k, d, q)-codes is equivalent to looking for solutions of the following diophantine (in-)equation system:

$$\begin{array}{rcl} M_q^k x & \leq & \begin{pmatrix} n-d \\ n-d \\ \vdots \\ n-d \end{pmatrix} \\ \mathbb{1}^T x & = & n \end{array}$$

where $x \in \mathbb{N}_0^m$ and M_q^k is the $m \times m$ incidence matrix between points (columns) and hyperplanes (rows) in $PG(k-1, q)$.

Problem: m gets huge very fast!

A possible approach:

A possible approach:

- Prescribe a subgroup A of $PGL(k, q)$ which must be contained in the automorphism group of the point (multi-)set P

A possible approach:

- Prescribe a subgroup A of $PGL(k, q)$ which must be contained in the automorphism group of the point (multi-)set P
- $a \in PGL(k, q)$ is automorphism of P iff $[p \in P \Leftrightarrow a(p) \in P]$

A possible approach:

- Prescribe a subgroup A of $PGL(k,q)$ which must be contained in the automorphism group of the point (multi-)set P
- $a \in PGL(k,q)$ is automorphism of P iff $[p \in P \Leftrightarrow a(p) \in P]$
- Instead of choosing single points of $PG(k-1,q)$, we now select complete orbits under the action of A on the points

A possible approach:

- Prescribe a subgroup A of $PGL(k, q)$ which must be contained in the automorphism group of the point (multi-)set P
- $a \in PGL(k, q)$ is automorphism of P iff $[p \in P \Leftrightarrow a(p) \in P]$
- Instead of choosing single points of $PG(k-1, q)$, we now select complete orbits under the action of A on the points $\Rightarrow$ number of variables is reduced to the number of orbits

A possible approach:

- Prescribe a subgroup A of $PGL(k, q)$ which must be contained in the automorphism group of the point (multi-)set P
- $a \in PGL(k, q)$ is automorphism of P iff $[p \in P \Leftrightarrow a(p) \in P]$
- Instead of choosing single points of $PG(k-1, q)$, we now select complete orbits under the action of A on the points $\Rightarrow$ number of variables is reduced to the number of orbits
- Let p be a point of $PG(k-1, q)$, H be a hyperplane and $a \in A$. Then we have: $p \in H \Leftrightarrow a(p) \in a(H)$.

A possible approach:

- Prescribe a subgroup A of $PGL(k,q)$ which must be contained in the automorphism group of the point (multi-)set P
- $a \in PGL(k,q)$ is automorphism of P iff $[p \in P \Leftrightarrow a(p) \in P]$
- Instead of choosing single points of $PG(k-1,q)$, we now select complete orbits under the action of A on the points
  $\Rightarrow$ number of variables is reduced to the number of orbits
- Let p be a point of $PG(k-1,q)$, H be a hyperplane and $a \in A$. Then we have: $p \in H \Leftrightarrow a(p) \in a(H)$.
  $\Rightarrow$ number of equations is reduced to the number of orbits of A on the hyperplanes.

Example (q=3, k=3)

	0	0	0	0	1	1	1	1	1	1	1	1	1
	0	1	1	1	0	0	0	1	1	1	2	2	2
	1	0	1	2	0	1	2	0	1	2	0	1	2
$(0\ 0\ 1)^{\perp}$	0	1	0	0	1	0	0	1	0	0	1	0	0
$(0\ 1\ 0)^{\perp}$	1	0	0	0	1	1	1	0	0	0	0	0	0
$(0\ 1\ 1)^{\perp}$	0	0	0	1	1	0	0	0	0	1	0	1	0
$(0\ 1\ 2)^{\perp}$	0	0	1	0	1	0	0	0	1	0	0	0	1
$(1\ 0\ 0)^{\perp}$	1	1	1	1	0	0	0	0	0	0	0	0	0
$(1\ 0\ 1)^{\perp}$	0	1	0	0	0	0	1	0	0	1	0	0	1
$(1\ 0\ 2)^{\perp}$	0	1	0	0	0	1	0	0	1	0	0	1	0
$(1\ 1\ 0)^{\perp}$	1	0	0	0	0	0	0	0	0	0	1	1	1
$(1\ 1\ 1)^{\perp}$	0	0	0	1	0	0	1	0	1	0	1	0	0
$(1\ 1\ 2)^{\perp}$	0	0	1	0	0	1	0	0	0	1	1	0	0
$(1\ 2\ 0)^{\perp}$	1	0	0	0	0	0	0	1	1	1	0	0	0
$(1\ 2\ 1)^{\perp}$	0	0	1	0	0	0	1	1	0	0	0	1	0
$(1\ 2\ 2)^{\perp}$	0	0	0	1	0	1	0	1	0	0	0	0	1
	1	1	1	1	1	1	1	1	1	1	1	1	1

Example (q=3, k=3)

$A = \langle \begin{pmatrix} 0 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix} \rangle$	0	0	0	0	1	1	1	1	1	1	1	1	1
	0	1	1	1	0	0	0	1	1	1	2	2	2
	1	0	1	2	0	1	2	0	1	2	0	1	2
$(0\ 0\ 1)^\perp$	0	1	0	0	1	0	0	1	0	0	1	0	0
$(0\ 1\ 0)^\perp$	1	0	0	0	1	1	1	0	0	0	0	0	0
$(0\ 1\ 1)^\perp$	0	0	0	1	1	0	0	0	0	1	0	1	0
$(0\ 1\ 2)^\perp$	0	0	1	0	1	0	0	0	1	0	0	0	1
$(1\ 0\ 0)^\perp$	1	1	1	1	0	0	0	0	0	0	0	0	0
$(1\ 0\ 1)^\perp$	0	1	0	0	0	0	1	0	0	1	0	0	1
$(1\ 0\ 2)^\perp$	0	1	0	0	0	1	0	0	1	0	0	1	0
$(1\ 1\ 0)^\perp$	1	0	0	0	0	0	0	0	0	0	1	1	1
$(1\ 1\ 1)^\perp$	0	0	0	1	0	0	1	0	1	0	1	0	0
$(1\ 1\ 2)^\perp$	0	0	1	0	0	1	0	0	0	1	1	0	0
$(1\ 2\ 0)^\perp$	1	0	0	0	0	0	0	1	1	1	0	0	0
$(1\ 2\ 1)^\perp$	0	0	1	0	0	0	1	1	0	0	0	1	0
$(1\ 2\ 2)^\perp$	0	0	0	1	0	1	0	1	0	0	0	0	1
	1	1	1	1	1	1	1	1	1	1	1	1	1

Example (q=3, k=3)

$A = \langle \begin{pmatrix} 0 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix} \rangle$	$\begin{matrix}0\\0\\1\end{matrix}$	$\begin{matrix}0\\1\\0\end{matrix}$	$\begin{matrix}0\\1\\1\end{matrix}$	$\begin{matrix}0\\1\\2\end{matrix}$	$\begin{matrix}1\\0\\0\end{matrix}$	$\begin{matrix}1\\0\\1\end{matrix}$	$\begin{matrix}1\\0\\2\end{matrix}$	$\begin{matrix}1\\1\\0\end{matrix}$	$\begin{matrix}1\\1\\1\end{matrix}$	$\begin{matrix}1\\1\\2\end{matrix}$	$\begin{matrix}1\\2\\0\end{matrix}$	$\begin{matrix}1\\2\\1\end{matrix}$	$\begin{matrix}1\\2\\2\end{matrix}$
$(0\ 0\ 1)^\perp$	0	1	0	0	1	0	0	1	0	0	1	0	0
$(0\ 1\ 0)^\perp$	1	0	0	0	1	1	1	0	0	0	0	0	0
$(0\ 1\ 1)^\perp$	0	0	0	1	1	0	0	0	0	1	0	1	0
$(0\ 1\ 2)^\perp$	0	0	1	0	1	0	0	0	1	0	0	0	1
$(1\ 0\ 0)^\perp$	1	1	1	1	0	0	0	0	0	0	0	0	0
$(1\ 0\ 1)^\perp$	0	1	0	0	0	0	1	0	0	1	0	0	1
$(1\ 0\ 2)^\perp$	0	1	0	0	0	1	0	0	1	0	0	1	0
$(1\ 1\ 0)^\perp$	1	0	0	0	0	0	0	0	0	0	1	1	1
$(1\ 1\ 1)^\perp$	0	0	0	1	0	0	1	0	1	0	1	0	0
$(1\ 1\ 2)^\perp$	0	0	1	0	0	1	0	0	0	1	1	0	0
$(1\ 2\ 0)^\perp$	1	0	0	0	0	0	0	1	1	1	0	0	0
$(1\ 2\ 1)^\perp$	0	0	1	0	0	0	1	1	0	0	0	1	0
$(1\ 2\ 2)^\perp$	0	0	0	1	0	1	0	1	0	0	0	0	1
	1	1	1	1	1	1	1	1	1	1	1	1	1

Example (q=3, k=3)

	$\begin{matrix}0&0&1\\0&1&0\\1&1&0\end{matrix}$	$\begin{matrix}0&1&1\\1&0&2\\0&2&2\end{matrix}$	$\begin{matrix}0&1&1\\1&0&1\\2&1&1\end{matrix}$	$\begin{matrix}1&1&1\\1&2&2\\0&0&1\end{matrix}$	$\begin{matrix}1\\1\\2\end{matrix}$
$(0\ 0\ 1)^\perp$	1	1	0	2	0
$(0\ 1\ 0)^\perp$	2	1	1	0	0
$(0\ 1\ 1)^\perp$	1	0	1	1	1
$(0\ 1\ 2)^\perp$	2	1	1	0	0
$(1\ 0\ 0)^\perp$	2	1	1	0	0
$(1\ 0\ 1)^\perp$	0	3	0	0	1
$(1\ 0\ 2)^\perp$	0	1	2	1	0
$(1\ 1\ 0)^\perp$	1	1	0	2	0
$(1\ 1\ 1)^\perp$	0	1	2	1	0
$(1\ 1\ 2)^\perp$	1	0	1	1	1
$(1\ 2\ 0)^\perp$	1	0	1	1	1
$(1\ 2\ 1)^\perp$	1	1	0	2	0
$(1\ 2\ 2)^\perp$	0	1	2	1	0
	3	3	3	3	1

Example (q=3, k=3)

	$\begin{matrix}0&0&1\\0&1&0\\1&1&0\end{matrix}$	$\begin{matrix}0&1&1\\1&0&2\\0&2&2\end{matrix}$	$\begin{matrix}0&1&1\\1&0&1\\2&1&1\end{matrix}$	$\begin{matrix}1&1&1\\1&2&2\\0&0&1\end{matrix}$	$\begin{matrix}1\\1\\2\end{matrix}$
$(0\ 0\ 1)^\perp$	1	1	0	2	0
$(0\ 1\ 0)^\perp$	2	1	1	0	0
$(0\ 1\ 1)^\perp$	1	0	1	1	1
$(0\ 1\ 2)^\perp$	2	1	1	0	0
$(1\ 0\ 0)^\perp$	2	1	1	0	0
$(1\ 0\ 1)^\perp$	0	3	0	0	1
$(1\ 0\ 2)^\perp$	0	1	2	1	0
$(1\ 1\ 0)^\perp$	1	1	0	2	0
$(1\ 1\ 1)^\perp$	0	1	2	1	0
$(1\ 1\ 2)^\perp$	1	0	1	1	1
$(1\ 2\ 0)^\perp$	1	0	1	1	1
$(1\ 2\ 1)^\perp$	1	1	0	2	0
$(1\ 2\ 2)^\perp$	0	1	2	1	0
	3	3	3	3	1

Example (q=3, k=3)

$(A^{-1}) = \langle \begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} \rangle$	$\begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{matrix}$	$\begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 2 \\ 0 & 2 & 2 \end{matrix}$	$\begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 2 & 1 & 1 \end{matrix}$	$\begin{matrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 0 & 0 & 1 \end{matrix}$	$\begin{matrix} 1 \\ 1 \\ 2 \end{matrix}$
$(0\ 0\ 1)^\perp$	1	1	0	2	0
$(0\ 1\ 0)^\perp$	2	1	1	0	0
$(0\ 1\ 1)^\perp$	1	0	1	1	1
$(0\ 1\ 2)^\perp$	2	1	1	0	0
$(1\ 0\ 0)^\perp$	2	1	1	0	0
$(1\ 0\ 1)^\perp$	0	3	0	0	1
$(1\ 0\ 2)^\perp$	0	1	2	1	0
$(1\ 1\ 0)^\perp$	1	1	0	2	0
$(1\ 1\ 1)^\perp$	0	1	2	1	0
$(1\ 1\ 2)^\perp$	1	0	1	1	1
$(1\ 2\ 0)^\perp$	1	0	1	1	1
$(1\ 2\ 1)^\perp$	1	1	0	2	0
$(1\ 2\ 2)^\perp$	0	1	2	1	0
	3	3	3	3	1

Example (q=3, k=3)

$(A^{-1}) = \langle \begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} \rangle$	$\begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{matrix}$	$\begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 2 \\ 0 & 2 & 2 \end{matrix}$	$\begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 2 & 1 & 1 \end{matrix}$	$\begin{matrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 0 & 0 & 1 \end{matrix}$	$\begin{matrix} 1 \\ 1 \\ 2 \end{matrix}$
$(0\ 0\ 1)^{\perp}$	1	1	0	2	0
$(0\ 1\ 0)^{\perp}$	2	1	1	0	0
$(0\ 1\ 1)^{\perp}$	1	0	1	1	1
$(0\ 1\ 2)^{\perp}$	2	1	1	0	0
$(1\ 0\ 0)^{\perp}$	2	1	1	0	0
$(1\ 0\ 1)^{\perp}$	0	3	0	0	1
$(1\ 0\ 2)^{\perp}$	0	1	2	1	0
$(1\ 1\ 0)^{\perp}$	1	1	0	2	0
$(1\ 1\ 1)^{\perp}$	0	1	2	1	0
$(1\ 1\ 2)^{\perp}$	1	0	1	1	1
$(1\ 2\ 0)^{\perp}$	1	0	1	1	1
$(1\ 2\ 1)^{\perp}$	1	1	0	2	0
$(1\ 2\ 2)^{\perp}$	0	1	2	1	0
	3	3	3	3	1

Example (q=3, k=3)

$(A^{-1}) = \langle \begin{pmatrix} 0&2&1 \\ 1&0&0 \\ 1&1&0 \end{pmatrix} \rangle$	$\begin{matrix} 0&0&1 \\ 0&1&0 \\ 1&1&0 \end{matrix}$	$\begin{matrix} 0&1&1 \\ 1&0&2 \\ 0&2&2 \end{matrix}$	$\begin{matrix} 0&1&1 \\ 1&0&1 \\ 2&1&1 \end{matrix}$	$\begin{matrix} 1&1&1 \\ 1&2&2 \\ 0&0&1 \end{matrix}$	$\begin{matrix} 1 \\ 1 \\ 2 \end{matrix}$
$(0\ 0\ 1)^{\perp}$	1	1	0	2	0
$(0\ 1\ 0)^{\perp}$	2	1	1	0	0
$(0\ 1\ 1)^{\perp}$	1	0	1	1	1
$(1\ 0\ 1)^{\perp}$	0	3	0	0	1
$(1\ 0\ 2)^{\perp}$	0	1	2	1	0
	3	3	3	3	1

The reduced system:

$$\begin{pmatrix} 1 & 1 & 0 & 2 & 0 \\ 2 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 3 & 0 & 0 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{pmatrix} \cdot x \leq \begin{pmatrix} n-d \\ n-d \\ n-d \\ n-d \\ n-d \end{pmatrix}$$

$$\begin{pmatrix} 3 & 3 & 3 & 3 & 1 \end{pmatrix} \cdot x = n$$

A heuristic solution algorithm:

A heuristic solution algorithm:

- Input: (In-)equation system of type

$$\begin{array}{rcll} A\,x & \leq & c & \text{(indices } 0 \ldots m-1) \\ B\,x & = & d & \text{(index } m)\,, \end{array}$$

with $x \in \mathbb{N}_0^n$, $A \in \mathbb{N}^{m \times n}$, $c \in \mathbb{N}^m$, $B \in \mathbb{N}_+^{1 \times n}$, $d \in \mathbb{N}_+$.

A heuristic solution algorithm:

- Input: (In-)equation system of type

$$\begin{array}{rcll} A\,x & \leq & c & (\text{indices } 0 \ldots m-1) \\ B\,x & = & d & (\text{index } m)\,, \end{array}$$

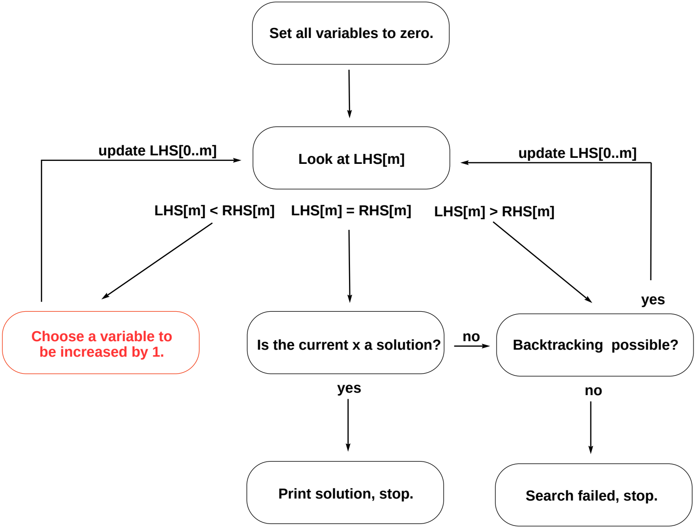
  with $x \in \mathbb{N}_0^n$, $A \in \mathbb{N}^{m \times n}$, $c \in \mathbb{N}^m$, $B \in \mathbb{N}_+^{1 \times n}$, $d \in \mathbb{N}_+$.
- Output: solution of the system or 'search failed'

A heuristic solution algorithm:

- Input: (In-)equation system of type

$$\begin{array}{rcll} A\,x & \leq & c & \text{(indices } 0 \ldots m-1\text{)} \\ B\,x & = & d & \text{(index } m\text{)}\,, \end{array}$$

  with $x \in \mathbb{N}_0^n$, $A \in \mathbb{N}^{m \times n}$, $c \in \mathbb{N}^m$, $B \in \mathbb{N}_+^{1 \times n}$, $d \in \mathbb{N}_+$.
- Output: solution of the system or 'search failed'
- Remark: algorithm can easily be generalized to other problems.


Set all variables to zero.
Look at LHS[m]
update LHS[0..m]
update LHS[0..m]
LHS[m] < RHS[m]
LHS[m] = RHS[m]
LHS[m] > RHS[m]
Choose a variable to be increased by 1.
Is the current x a solution?
no
Backtracking possible?
yes
yes
no
Print solution, stop.
Search failed, stop.

Choice of the variable to be increased next:

Choice of the variable to be increased next:

- For each variable v do the following:

Choice of the variable to be increased next:

- For each variable v do the following:
    - compute left hand sides after increase of v and store it in initialLHS[0..m]

Choice of the variable to be increased next:

- For each variable v do the following:
  - compute left hand sides after increase of v and store it in initialLHS[0..m]
  - set $counter[0] = counter[1] = \ldots = counter[m-1] = 0$

Choice of the variable to be increased next:

- For each variable v do the following:
  - compute left hand sides after increase of v and store it in initialLHS[0..m]
  - set $counter[0] = counter[1] = \ldots = counter[m-1] = 0$
  - do n_s sample runs (n_s being a number fixed by the user)

Choice of the variable to be increased next:

- For each variable v do the following:
  - compute left hand sides after increase of v and store it in initialLHS[0..m]
  - set $counter[0] = counter[1] = \ldots = counter[m-1] = 0$
  - do n_s sample runs (n_s being a number fixed by the user)
  - set $eval(v) := \prod_{j=0}^{m-1} \frac{\text{counter[j]}}{n_s}$

Choice of the variable to be increased next:

- For each variable v do the following:
  - compute left hand sides after increase of v and store it in initialLHS[0..m]
  - set $counter[0] = counter[1] = \ldots = counter[m-1] = 0$
  - do n_s sample runs (n_s being a number fixed by the user)
  - set $eval(v) := \prod_{j=0}^{m-1} \frac{\text{counter[j]}}{n_s}$
- choose v^* so that $eval(v^*)$ is maximal

Pseudocode for a single sample run:

```
for (int i=0; i<=m; i++){ //restore initial LHS
    LHS[i]=initialLHS[i];
}
while(LHS[m]<RHS[m]){ //increase vars randomly
    randomly choose a variable w;
    increase w by 1;
    update LHS[0],LHS[1],...,LHS[m-1],LHS[m];
}
if (LHS[m]==RHS[m]){ //update counters
    for (i=0; i<m; i++){
        if (LHS[i]<=RHS[i]){
            counter[i]++;
        }
    }
}
```

With the method presented we could construct the following new linear binary codes:

$k = 11:$

n	d
41	**16**
73	**32**
81	34
136	62
139	**64**
146	66
149	68
155	**72**

$k = 12:$

n	d
74	**32**
83	34
99	42
102	44
107	46
110	48
140	**64**

$k = 13:$

n	d
41	**14**
155	68
158	70
161	72

(entries in boldface belong to optimal codes)

Thanks for your attention!