International Journal of Mathematics and Statistics Invention (IJMSI)
E-ISSN: 2321 – 4767 P-ISSN: 2321 - 4759
www.ijmsi.org Volume 12 Issue 4 || Jul.-Aug. 2024 || PP-08-14

A Novel Construction Algorithm for Some Optimal Binary Linear Codes

Skezeer John B. Paz[1] and Ederlina G. Nocon[2]
[1]*Department of Natural Sciences and Mathematics, Notre Dame of Marbel University, Alunan Avenue, City of Koronadal, South Cotabato, Philippines*
[2]*Department of Mathematics and Statistics, De La Salle University, 2401 Taft Ave., Malate, Manila, Philippines*

ABSTRACT*: One of the main challenges in coding theory is finding optimal binary linear codes. A binary linear code* $C = [n, k, d]$ *is considered optimal if no other linear code with the same length* n *and dimension* k *has a greater minimum distance* d*. Theoretical bounds provide limits for the minimum distance* d *of a linear code given* n *and* k*. For instance, the lower bound indicates that a linear code with a specified minimum distance exists, while the upper bound can be determined using results such as the Griesmer bound. Achieving an optimal binary linear code involves constructing a code where the lower and upper bounds of* d *coincide. In previously published tables, Andries Brouwer presented some optimal binary linear codes for* $1 \leq n \leq 256$ *and* $k \leq n$*. Markus Grassl further improved these results by detailing construction processes for codes achieving the lower bound. In this paper, we introduce new optimal binary linear codes using novel construction methods applied to existing codes. Specifically, we developed a construction algorithm that extends the list of optimal binary linear codes for* $257 \leq n \leq 300$ *with* $k \leq 7$*.*

KEYWORDS*: Coding Theory, Binary Linear Codes, Griesmer Bound, Construction of Optimal Binary Linear Codes*

Date of Submission: 01-07-2024 Date of Acceptance: 12-07-2024

I. INTRODUCTION

Coding theory is the study of methods for the efficient and accurate transfer of information from one party to another. The physical medium through which the information is transmitted is called a channel. Some examples of channels are network communication, USB channel, satellite communication, and so on. It is worth noting that all communication channels are susceptible to errors. Often, undesirable disturbances may result in the distortion of transmitted information, and these disturbances are referred to as noise. In this regard, codes were developed to correct errors on noisy and inaccurate communication channels. Thus, coding theory deals with the problem of detecting and correcting transmission errors caused by noise in the channel. Encoding is the process of adding redundancy digits to the information digits while decoding is the process of getting the information being sent. Hence, coding theory poses two main questions: (1) to construct codes that can correct a maximal number of errors while using a minimal amount of redundancy, and (2) to construct codes with efficient encoding and decoding procedures.

On one hand, linear codes are some of the most basic codes in coding theory. A linear code C with length n over $\mathbb{F}_q$ is a vector subspace of $\mathbb{F}_q^n$, where $\mathbb{F}_q$ is a field of order q. Linear codes are simple to understand but are very important and effective for practical applications, such as encoding and decoding messages sent over communication channels. Since coding theory is interested in constructing codes that can correct a maximal number of errors while using a minimal amount of redundancy, finding optimal linear codes had been one of its central problems. A linear code $C = [n, k, d]$ is called optimal if there is no linear code with a higher minimum distance d given the length n and the dimension k. For the purpose of error-correcting codes, we are interested in codes with high minimum distance d as these allow the correction of $\lfloor (d-1)/2 \rfloor$ errors [6]. For the purpose of discussion, any mention of codes in this paper refers to linear codes, unless otherwise stated.

There are several methods of constructing new linear codes from a given linear code. These include lengthening, extending, subcoding, shortening, puncturing, juxtaposition, combination, and Plotkinsum construction. A combination of these methods results in another linear code. One way to find an optimal binary linear code is to construct a binary linear code C that achieves the highest possible value of its minimum distance d, given length n and dimension k. That is, constructing a binary linear code which results in the code $C = [n, k, d]$, where d is the upper bound of the minimum distance. We note that the lower bound and the upper bound of d do not always meet. The upper bound may be taken by using some theoretic results such as the Griesmer Bound [2], and the lower bound may be taken by constructing a linear code with the highest distance d, given n and k.

Meanwhile, a binary linear code is a linear code over a field $\mathbb{F}_2$. Andries Brouwer [1] published a table of the lower bounds and upper bounds of the minimum distance d of binary linear codes for $1 \leq n \leq 256$ and $k \leq n$. In addition to the bounds, some information on the construction of the codes and some references were provided. Markus Grassl [3] further improved this table by increasing the values of some lower bounds and constructing the binary linear codes exhibiting these lower bounds. In such table, in most cases, for $n > 30$ and $k > 7$, the upper bounds and lower bounds of the values of d are not equal. This is due to the fact that the constructed binary linear codes with the highest d (the lower bounds) do not reach the computed upper bounds of the minimum distance d. In such cases, the constructed binary linear codes are called "good codes." On the other hand, in most cases, for $k \leq 7$ the lower bounds of d are equal to its higher bounds, making the constructed binary linear codes optimal.

In this paper, we are interested in finding new optimal binary linear codes. We do this by getting the upper bounds of the minimum distance of the codes using the Griesmer bound and constructing binary linear codes which meet the upper bound of d, given the values of n and k. Particularly, since we wanted to construct optimal binary linear codes, we focus on constructing optimal binary linear codes for $257 \leq n \leq 300$ and $k \leq 7$. We do this by applying a construction algorithm on the codes already constructed by Grassl and on other codes that we had already constructed.

II. LINEAR CODES

In this section, we discuss some basic concepts in coding theory such as the basic definitions, examples, lemmas and theorems; the Griesmer bound which gives the upper bound of a parameter of a binary linear code; and some methods on constructing new binary linear codes from existing binary linear codes.

2.1. Basic Concepts

A *linear code* $C = [n, k, d]$ with length n, dimension k and minimum distance d, over a field $\mathbb{F}_q$ is a vector subspace of $\mathbb{F}_q^n$. If $\mathbb{F}_q = \{x_1, x_2, \dots, x_q\}$ is a field of size q, we call $\mathbb{F}_q$ an *alphabet* and the x_i values as *symbols*. A *block code* C of length n over $\mathbb{F}_q$ is a nonempty subset of $\mathbb{F}_q^n$. A vector $x \in C$ is called a *codeword* and the number of codewords in C, denoted by $|C|$, is called the *size* of C. A code of length n and size M is called (n, M)*-code*. The dimension k of the linear code C is the dimension of C as a vector subspace of $\mathbb{F}_q^n$. We begin to use $\mathbb{F}_2$ or $GF(2)$, the Galois Field of order 2, in preference to $\{0, 1\}$ to denote our binary alphabet, since we wish to emphasize that the alphabet carries with it an arithmetic structure. The *minimum distance* of a code C, denoted by $\mathrm{d}(C)$, is defined by

$$\mathrm{d}(C) = \min\{\mathrm{d}(x, y) : x, y \in C, x \neq y\}.$$

If C has length n, dimension k and minimum distance $\mathrm{d}(C) = d$, then C is referred to as $[n, k, d]$*-code*. The numbers n, k and d are called the *parameters* of the linear code. The $k \times n$ matrix G whose rows form a basis of an $[n, k]$-code C is called a *generator matrix* of C. The *hamming weight* of x, denoted by $\mathrm{wt}(x)$, for $x \in \mathbb{F}_q^n$, is defined to be the number of nonzero coordinates of x. That is,

$$\mathrm{wt}(x) = |\{i : x_i \neq 0\}|.$$

The *minimum weight* of a linear code C, denoted by $\mathrm{wt}(C)$, is the smallest among the weights of all nonzero codewords in C. We note that for a linear code C over $\mathbb{F}_q^n$, the minimum distance $\mathrm{d}(C)$ is equal to minimum weight $\mathrm{wt}(C)$. Also, for a linear code C of length n and dimension k, over $\mathbb{F}_q$, the number of codewords in C is equal to q^k. This follows from the fact that each codeword of C can be expressed uniquely as a linear combination of the basis vectors (a subspace with dimension k has q^k elements). We can clearly see that a binary linear code C of dimension k contains precisely 2^k codewords.

A *trivial linear code* is a linear code with parameter $[n, 1, n]$ or $[n, n, 1]$, for all $n \in \mathbb{N}$. The $[n, 1, n]$-linear code is a code with one non-zero codeword, and the weight of this codeword is n. The $[n, n, 1]$-linear code is a code that contains the whole subspace $\mathbb{F}_q^n$, with a generator matrix row equivalent to an identity matrix, I_n, and has minimum weight of 1. The *repetition codes* are codes consisting of codewords $x_i, x_i, \dots, x_i$ for $x_i \in \mathbb{F}$. A *cyclic code* is a linear code whose any cyclic shift of a codeword is also a codeword, i.e., whenever $x_0, \dots, x_{n-1} \in C$, then also $x_{n-1}, x_0, \dots, x_{n-2} \in C$. On the other hand, a *quasi-cyclic* code is an (n, k) linear block code of dimensions $n = mn_o$ and $k = mk_o$ where every cyclic shift of a codeword by no symbols yields another codeword.

For a linear code C over $\mathbb{F}_q^n$, the *dual code* of C, denoted by $C^\perp$, is the orthogonal complement of subspace C, that is,

$$C^\perp = \{x \in \mathbb{F}^n : x \cdot c = 0, \forall c \in C\}.$$

where $x \cdot c$ is the usual dot product. We note that $(C^\perp)^\perp = C^{\perp\perp} \supseteq C$ and $\mathrm{C}^{\perp\perp}$=C. If C is an $[n, k]$ linear code over $\mathbb{F}$, then its dual $C^\perp$ is an $[n, n-k]$ linear code over $\mathbb{F}$. The linear code C is *self-orthogonal* if $C \subseteq C^\perp$ and is *self-dual* if $C = C^\perp$. A generator matrix H for the dual code $C^\perp$ of the linear code C is sometimes called a check matrix for C. A *parity check matrix* for C is an $(n-k) \times n$ matrix H such that $x \in C$ if and only if $xH^T = 0$.

2.2. The Griesmer Bound

The following is a theorem developed by James Hugo Griesmer [5], that can be used to find the upper bound of the minimum distance d of a (binary) linear code $C = [n, k, d]$.

Theorem 1 (Griesmer Bound) *Let C be an $[n, k, d]$ code over $\mathbb{F}_q$ with $k \geq 1$. Then,*

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil \tag{1}$$

2.3. Existing Methods in Constructing Codes from Other Codes

We now discuss a few methods in constructing binary linear codes from old binary linear codes such as lengthening, extending, subcoding, shortening, puncturing, juxtaposition, combination, and Plotkinsum construction. We also give the theorems that are used in identifying the changes in the parameters of the new code resulting from the constructions described. We refer the readers to [6], [8] and [9] for the detailed proofs of these.

2.3.1. Lengthening a Code

For a binary linear code C, the process of adding 0 to all codewords in C is referred to as lengthening the code C.

Theorem 2. *Let C be an $[n, k, d]$ binary linear code. Then, there exists a linear code C^* with parameter $[n + s, k, d]$, where $s > 0$.*

2.3.2. Extending a Code

For a binary linear code C of length n, the code C^* of length $n + 1$ obtained from C by adding one extra bit to each codeword in order to make each codeword into a new codeword of even weight is called the extended code of C. In particular, we are interested in codes that have codewords with odd weight initially.

Theorem 3. *Let C be an $[n, k, d]$ binary linear code. Then, given that d is odd, there exists a binary linear code C^* with parameter $[n + 1, k, d + 1]$.*

2.3.3. Taking a Subcode from a Code

For a binary linear code C of length n, the process of deleting a codeword from the basis of C so as to obtain a new code C^* where the minimum weight of C^* remains the same, is referred to as taking a subcode of C, or subcoding the code C.

Theorem 4. *Let C be an $[n, k, d]$ binary linear code. Then, there exists a binary linear code C^* with parameter $[n, k - s, d]$, where $s > 0$ and $k > 1$.*

2.3.4. Shortening a Code

For a binary linear code C of length n, a shortened code of C is the set of all codewords of C which are zero at a fixed coordinate with that coordinate deleted. Those non-zero codewords in C shall be removed from C. In particular, we are interested in deleting those coordinates that are not 0 for all codewords in C. If the coordinate is 0 for all codewords in C, we shall use the reverse technique of lengthening, and then use the technique of taking a subcode.

Theorem 5. *Let C be an $[n, k, d]$ binary linear code. Then, there exists a binary linear code C^* with parameter $[n - s, k - s, d]$, where $k > s > 0$.*

2.3.5. Puncturing or Truncating a Code

For a binary linear code C of length n, the code obtained by removing a fixed coordinate of C is called a punctured code of C. In particular, we are interested in deleting those coordinates that is not 0 for all codewords in C.

Theorem 6. *Let C be an $[n, k, d]$ binary linear code. Then, there exists a binary linear code C^* with parameter $[n - s, k, d - s]$, where $s > 0$ and $d > 1$.*

2.3.6. Juxtaposition or Concatenation of Codes

The concatenation of two codewords, c_1 and c_2, is a new string c_1c_2 formed by writing the elements of c_1 and the elements of c_2 consecutively. The concatenation of two codes, C_1 and C_2, is a set of codewords of the form c_1c_2 such that $c_1 \in C_1$ and $c_2 \in C_2$. In particular, the sizes of C_1 and C_2 are both equal. We define concatenation as an injective operation. That is, each codewords in C_1 and C_2 can only be used once in the concatenation. Then, the new code has a generator matrix $G^* = (G_1G_2)$, where G_1 and G_2 are the generator matrices of C_1 and C_2, respectively.

Theorem 7. *Let C_1 be an $[n_1, k, d_1]$ binary linear code and C_2 be an $[n_2, k, d_2]$ binary linear code. Then, there*

exists a binary linear code C^* *with parameter* $[n_1 + n_2, k, d_1 + d_2]$.

2.3.7. Combination of Codes

The combination of two codes, C_1 and C_2, is a set of codewords of the form c_1c_2 such that $c_1 \in C_1$ and $c_2 \in C_2$. We define combination as a surjective operation, which is different from concatenation of two codes. That is, every codewords in C_1 will concatenate with every codewords in C_2. In particular, the sizes of C_1 and C_2 need not to be equal.

Theorem 8. *Let* C_1 *be an* $[n_1, k_1, d_1]$ *binary linear code and* C_2 *be an* $[n_2, k_2, d_2]$ *binary linear code. Then, there exists a binary linear code* C^* *with parameter* $[n_1 + n_2, k_1 + k_2, min\{d_1, d_2\}]$.

2.3.8. Plotkinsum or $u(u + v)$-Construction

Plotkinsum, also called $u(u + v)$ construction and bar product, is a classic tool to construct new binary linear codes from a binary linear code already known. Particularly, we consider two binary linear codes of the same length and of which one code has minimum distance which is twice the minimum distance of the other.

Theorem 9. *Let* C_1 *be an* $[n, k_1, d]$ *binary linear code and* C_2 *be an* $[n, k_2, 2d]$ *binary linear code. Let* $C_1 \oplus C_2$ *be the code consisting of all codewords of the form*

$$(u, u + v) = (u_1, u_2, \dots, u_n, u_1 + v_1, u_2 + v_2, \dots, u_n + v_n)$$

with

$$u = (u_1, u_2, \dots, u_n) \in C_1 \text{ and } v = (v_1, v_2, \dots, v_n) \in C_2.$$

Then, $C_1 \oplus C_2$ *is a* $[2n, k_1 + k_2, 2d]$ *binary linear code.*

III. CONSTRUCTING OPTIMAL BINARY LINEAR CODES

Andries Brouwer [1] published a table of the lower bounds and upper bounds of the minimum distance d of binary linear codes for $1 \leq n \leq 256$ and $k \leq n$, which was further improved by Markus Grassl [3] by increasing the values of some lower bounds and constructing the binary linear codes exhibiting these lower bounds by using some known binary linear codes. Significantly, some optimal binary linear codes were constructed for those of which the lower bounds were to equal the higher bounds (most of which are those codes with $k \leq 7$). In this section, we extend the list of these optimal binary linear codes up to $257 \leq n \leq 300$ and $k \leq 7$. Particularly, we use the Griesmer Bound in taking the upper bound of each minimum distance of the codes and we apply a construction algorithm on the codes already constructed by Grassl in taking the lower bounds.

3.1. Taking the Upper Bounds Using the Griesmer Bound Theorem

We now show a process of taking the upper bounds of the minimum distance *d* of the binary linear codes for $257 \leq n \leq 300$ and $k \leq 7$, using the Griesmer Bound.

Let us suppose that the binary linear code $[257, 2, 172]$, where $n = 257$, $k = 2$ and $d = 172$, exists. Then, equation 1 has the following representation:

$$257 \geq \sum_{i=0}^{1} \left\lceil \frac{172}{2^i} \right\rceil$$

so that $257 \ngeq 258$.

By Theorem 1, this implies that the binary linear code [257,2,172] does not exist. However, if we replace $d = 172$ by $d = 171$, we have the binary linear code $[257, 2, 171]$. Then, equation 1 has the following representation:

$$257 \geq \sum_{i=0}^{1} \left\lceil \frac{171}{2^i} \right\rceil$$

so that $257 \geq 257$. Thus, the upper bound of d for a binary linear code with $n = 257$ and $k = 2$ is 171 (*as shown in row 1 column 2 of Table 2*). We use similar process in getting other values of the upper bounds of *d* for $257 \leq n \leq 300$ and $k \leq 7$.

3.2. Constructing Codes Exhibiting the Lower Bounds

We now present a construction algorithm that we shall use in constructing a binary linear code exhibiting the lower bound of the minimum distance d, for a given value of length n and dimension k.

In constructing the binary linear code $C = [n, k, d]$, with length n, dimension k, and minimum distance upper bound d, we follow the algorithm as shown in Table 1. Please note that the (optimal) binary linear codes C^*, C_1^* and C_2^* that we look for in the algorithm are the codes already constructed by Markus Grassl [4] and/or the codes that we had already constructed.

Table 1. A construction algorithm for optimal binary linear codes.

Algorithm

Given $C = [n, k, d]$,
if $k = 1$,
  then C is a trivial binary linear code.
else
  if d is odd,
  look for ${C_1}^* = [n_1, k, d_1]$ and ${C_2}^* = [n_2, k, d_2]$ such that $n = n_1 + n_2$ and $d = d_1 + d_2$
  then apply juxtaposition to ${C_1}^*$ and ${C_2}^*$.
  else
  look for the existence of $C^* = [n-1, k, d]$
  if it exists,
  then apply lengthening to C^* by 1.
  else
  look for $C^* = [n-1, k, d-1]$
  then apply extending to C^* by 1.

Suppose we construct the binary linear code [258, 6 ,129], where length $n = 258$, dimension $k = 6$, and minimum distance upper bound $d = 129$. Since d is odd, we look for the binary linear codes of parameters $C_1^* = [n_1, k, d_1]$ and $C_2^* = [n_2, k, d_2]$ such that $n = n_1 + n_2$ and $d = d_1 + d_2$ in the list of binary linear codes already constructed (*refer to Appendix B*). Now, since the binary linear codes $[252, 6, 128]$ and $[6, 6, 1]$ exist, we apply juxtaposition to [252,6,128] and [6, 6, 1]. We now get the optimal binary linear code $[258, 6, 129]$. Table 2 shows the complete list of optimal binary linear codes (colored red) constructed through this method.

Now, suppose we construct the binary linear code [257,6,128], where length $n = 257$, dimension $k = 6$, and minimum distance upper bound $d = 128$. Since d is even, we consider looking for either binary linear codes of parameters $[256, 6, 128]$ or $[256, 6, 127]$ in the list of binary linear codes already constructed (*refer to Appendix B*). Now, since the binary linear code $[256, 6, 128]$ exists, we apply lengthening to $[256, 6, 128]$ by 1. We now get the optimal binary linear code $[257, 6, 128]$. Table 2 shows the complete list of optimal binary linear codes (colored violet) constructed through this method.

On the other hand, suppose we construct the binary linear code [257,5,132], where length $n = 257$, dimension $k = 5$, and minimum distance upper bound $d = 132$. Since d is even, we consider looking for either binary linear codes of parameters [256,6,128] or $[256, 6, 127]$ in the list of binary linear codes already constructed (*refer to Appendix B*). Now, since the binary linear code $[256, 5, 131]$ exists, we apply extending to $[256, 5, 131]$ by 1. We now get the optimal binary linear code $[257, 5, 132]$. Table 2 shows the complete list of optimal binary linear codes (colored green) constructed through this method. A trivial binary linear code $[n, 1, n]$ (the code with one non-zero codeword) can easily be constructed.

In Table 2, we summarize the optimal binary linear codes for $257 \leq n \leq 300$ and $k \leq 7$. Please note that the color of each entry in the minimum distance corresponds to the following construction: trivial construction - *black*; construction by applying juxtaposition - *red*; construction by applying lengthening - *violet*; construction by applying extending - *green*.

IV. CONCLUSIONS

Construction of optimal binary linear codes is a central problem in coding theory. Andries Brouwer published a table of the lower bounds and upper bounds of the minimum distance d of binary linear codes for $1 \leq n \leq 256$ and $k \leq n$ which was further developed by Markus Grassl. In his table, Grassl was able to improve the lower bounds of the codes presented by Brouwer and presented detailed construction for each code exhibiting the lower bounds. Some optimal binary linear codes were then constructed for those results in which the upper bounds and lower bounds of d are equal. To improve the list of the optimal binary linear codes which were already constructed by Grassl, we developed an algorithm applied to (optimal) binary linear codes already constructed to construct optimal binary linear codes for $257 \leq n \leq 300$ and $k \leq 7$. Particularly, we were able to construct the optimal binary linear codes by: (1) taking the upper bounds of the minimum distance d of the binary linear codes using Griesmer Bound; and (2) constructing binary linear codes which satisfy these upper bounds. More so, we were able to show a detailed construction method for each (optimal) binary linear code exhibiting the minimum distance lower bound.

Table 2. Optimal binary linear codes for $257 \leq n \leq 300$ and $k \leq 7$.

$n\backslash k$	1	2	3	4	5	6	7
257	257	171	146	136	132	128	128
258	258	172	147	136	132	129	128
259	259	172	148	137	132	130	128
260	260	173	148	138	133	130	128
261	261	174	148	138	134	131	129
262	262	174	149	139	134	132	130
263	263	175	150	140	135	132	130
264	264	176	150	140	136	132	131
265	265	176	151	140	136	133	132
266	266	177	152	141	136	134	132
267	267	178	152	142	136	134	132
268	268	178	152	142	137	135	133
269	269	179	153	143	138	136	134
270	270	180	154	144	138	136	134
271	271	180	154	144	139	136	135
272	272	181	155	144	140	136	136
273	273	182	156	144	140	137	136
274	274	182	156	145	140	138	136
275	275	183	156	146	141	138	136
276	276	184	157	146	142	139	137
277	277	184	158	147	142	140	138
278	278	185	158	148	143	140	138
279	279	186	159	148	144	140	139
280	280	186	160	148	144	141	140
281	281	187	160	149	144	142	140
282	282	188	160	150	144	142	140
283	283	188	161	150	144	143	141
284	284	189	162	151	145	144	142
285	285	190	162	152	146	144	142
286	286	190	163	152	146	144	143
287	287	191	164	152	147	144	144
288	288	192	164	152	148	144	144
289	289	192	164	153	148	145	144
290	290	193	165	154	148	146	144
291	291	194	166	154	149	146	144
292	292	194	166	155	150	147	145
293	293	195	167	156	150	148	146
294	294	196	168	156	151	148	146
295	295	196	168	156	152	148	147
296	296	197	168	157	152	149	148
297	297	198	169	158	152	150	148
298	298	198	170	158	152	150	148
299	299	199	170	159	153	151	149
300	300	200	171	160	154	152	150

Meanwhile, since this paper only focused on the construction of optimal binary linear codes for $257 \leq n \leq 300$ and $k \leq 7$, further studies may consider increasing the parameters n and k, with $k \leq n$. Further studies may also deal on constructing optimal linear codes over other fields. More so, in most cases, for binary linear codes with $n > 30$ and $8 \leq k \leq n$, the lower bounds of d found are less than the upper bounds of d. It is of great

interest to construct those codes which satisfy the upper bounds of d. It would even be a great discovery if we can develop an algorithm in finding optimal (binary) linear codes for greater values of n and/or k.

ACKNOWLEDGMENTS

The authors express their heartfelt gratitude to DOST-SEI Accelerated Science and Technology Human Resource Development Program (ASTHRDP), Notre Dame of Marbel University, and De La Salle University for their research assistance, and Dr. Markus Grassl for unselfishly sharing their knowledge related to this study.

REFERENCES

[1] Brouwer, A., *Bounds on Linear Codes*, Vera S. Pless and W. Cary Huffman (Eds.), Handbook of Coding Theory, pp. 295-461, Elsevier, 1998.
[2] W. O. Alltop, *A Method for Extending Binary Linear Codes*, IEEE Transactions on Information Theory 30 (1984), 871–872.
[3] Grassl, M., *Searching for Linear Codes with Large Minimum Distance*, Wieb Bosma and John Cannon (Eds.) Discovering Mathematics with Magma. Springer, 2006.
[4] Grassl, Markus. *Bounds on the minimum distance of linear codes and quantum codes*, Online available at http://www.codetables.de.
[5] Griesmer, J. H. *A bound for Error-correcting Codes*, IBM Journal of Res. and Dev., vol. 4, no. 5, pp. 532-542, 1960.
[6] Tilborg, H. (1993), *Coding Theory: A First Course*, Eindhoven, Netherlands.
[7] Braun, M., Kohnert, A., & Wassermann, A., *Construction of linear codes with prescribed minimum distance*, University of Bayreuth, D-95440 Bayreuth, Germany.
[8] Lindell, Y. (2010), *Introduction to Coding Theory Lecture Notes*, Department of Computer Science Bar-Ilan University, Israel.
[9] Soh Joo Kiat, K., *Construction of Binary Linear Codes*, National University of Singapore, 2000.
[10] Ling, S. & Xing, C. (2004), *Coding Theory / A First Course*, Cambridge University Press.