Optimization of Non Binary Parity Check Coefficients

Emmanuel Boutillon

Optimization of Non Binary Parity Check Coefficients

Emmanuel Boutillon, *Senior Member, IEEE*

Lab-STICC, UMR 6582, Université de Bretagne Sud 56100 Lorient, France

email: emmanuel.boutillon@univ-ubs.fr

Abstract

This paper generalizes the method proposed by Pouillat et al. for the determination of the optimal Galois Field coefficients of a Non-Binary LDPC parity check constraint based on the binary image of the code. Optimal, or almost-optimal, parity check coefficients are given for check degree varying from 4 to 20 and Galois Field varying from GF(64) up to GF(1024). For all given sets of coefficients, no codeword of Hamming weight two exists. A reduced complexity algorithm to compute the binary Hamming weight 3 of a parity check is proposed. When the number of sets of coefficients is too high for an exhaustive search and evaluation, a local greedy search is performed. Explicit tables of coefficients are given. The proposed sets of coefficients can effectively replace the random selection of coefficients often used in NB-LDPC construction.

Index Terms

Non-Binary Parity Check, Non-Binary LDPC, Hamming Weight, Error control code

.

I. Introduction

Non-Binary Low Density Parity Check Codes (NB-LDPC) have been proposed by Mackay and Neal in 1996 as a generalization of the LPDC matrices [1]. In [2], Pouillat et al. present in 2008 a method to set the non-zero coefficients of a non binary parity check matrix H. The first step of the method concerns the problem of row optimization, i.e, the selection of the coefficients associated to a given parity check. The principle is to optimize the Hamming weight spectrum of the binary code $(md_c, m(d_c-1))$ associated to a parity check of degree d_c over a Galois Field GF(q) with $m = \log_2(q)$. The authors show that the higher the minimum distance of the binary

equivalent code, the better is the convergence of the NB-LDPC code in the waterfall region. They also show that, for two parity checks with the same associated binary minimum distance d_H, the multiplicity of binary codewords of Hamming distance d_H verifying the parity check equation should be minimized. Once the coefficients of the parity check equation are selected, the second step of [2] is to enumerate the cycles of short lengths in the Tanner graph associated to the parity check matrix and to constraint the GF(q) coefficients associated to each cycle so that only the zero codeword is associated to the short length cycles. This second step is out of the scope of this paper. The state of the art on coefficient selection is quite sparse, except in [3] and [4]. In [3], Mackay proposed to select the set of non null coefficients that maximizes the marginal entropy of one element of the syndrome vector. In [4], a method used to construct the NB-LDPC code used by the Consultative Committee for Space Data Systems (CCSDS) are presented and some sets of coefficients for $d_c = 4$ over GF(64) and GF(256) are given. We should also mention the paper of [5] which shows minimum Hamming distance upper bound of short length binary codes.

A direct exploration of all possible codewords associated to a given set of coefficients is limited to small check node degree and Galois Field order due to the exponential increase of complexity. In fact, the number of codewords for a parity check of degree d_c over GF(q) is q^{d_c-1}. For example, for $d_c = 5$ over GF(64), there is $64^4 = 16.8 \times 10^6$ codewords per set of coefficients. The number of sets of coefficients is around 8×10^4 (see section III): the direct method shows rapidly its limit since it requires more that 100 billions of operations. In [2], optimal, or almost optimal, sets of coefficients are only given for $d_c = 4$ over GF(64), GF(128) and GF(256).

In this paper, we revisit the problem of coefficient optimization in the case where the binary hamming weight associated to the parity check is equal to 3. We propose a method with a complexity of $\mathcal{O}(d_c^2)$ to evaluate the number of codewords of weight 3. When the number of sets of coefficients is too high for an exhaustive search, a local greedy search is performed. Explicit tables of coefficients are given for d_c varying from 3 to 20 and for Galois Field GF(64) up to GF(1024). The proposed sets of coefficients can effectively replace the random selection of coefficients often used in NB-LDPC construction. For example, let us consider a check node of degree 12 over GF(256), then, in average, randomly selected set of coefficients leads to 68 codewords of weight 3 while the optimized set of coefficients has only 11 codewords of weight 3. In other words, using proposed coefficients, each parity check equation has a better individual error correction, leading globally to a better convergence in the waterfall of the whole NB-LDPC

code.

The remaining of the paper is organized as follows. Section II presents the background on parity check equation and states the optimization problem. Section III evaluates the number of configurations to be evaluated. Section IV proposes an heuristic method to determine the effective set of coefficients. Finally, section IV concludes the paper. All the sets of optimal/optimized coefficients are given in the annex.

II. Determination of optimal coefficients of a Parity Check equation

The Galois Field GF(2^m) will be represented by the set of polynomials over GF(2) modulo $P_m[X]$, where $P_m[X]$ is an irreducible polynomial of degree m. Thus, by definition, GF(2^m) = GF(2)$[X]/P_m[X]$. It is usual to represent the element of this field either by setting $X = \alpha$ and representing the non null element as power of α, i.e, if $x \in$ GF(q), then $x \neq 0$ implies that x can be written as $x = \alpha^a$, with a a natural that takes its value between 0 and $q-2$. It is also possible to represent the element of GF by a binary vector of size m that represents the coefficients of a polynomial of GF(2)$[X]/P_m[X]$ over the base $(1, \alpha, \ldots \alpha^{m-1})$. In this paper, we use the following irreducible polynomials to construct the Galois Field of size 32 up to 1024.

$$\begin{cases} P_5[X] &= 1 + X^2 + X^5 \\ P_6[X] &= 1 + X + X^6 \\ P_7[X] &= 1 + X^3 + X^7 \\ P_8[X] &= 1 + X^2 + X^3 + X^4 + X^8 \\ P_9[X] &= 1 + X^5 + X^9 \\ P_{10}[X] &= 1 + X^4 + X^{10} \end{cases} \tag{1}$$

A parity check code $\mathcal{C}$ of degree d_c over GF(q)^{d_c} is a code defined by a set of d_c non-null GF(q) coefficients $H = \{h_i\}_{i=1,2,\ldots d_c}$, with $h_i = \alpha^{a_i}$. Vector $\mathbf{X} = (x_1, x_2, \ldots, x_{d_c})$ of GF(q)^{d_c} belongs to the code $\mathcal{C}$ if and only if

$$h_1 x_1 + h_2 x_2 + \ldots + h_{d_c} x_{d_c} = 0, \tag{2}$$

where additions and multiplications are done in GF(q). Since addition in GF is commutative, the order of the coefficients doesn't impact the properties of the code. Moreover, multiplying (2) by a constant factor doesn't change the equation. In other words, we can always select the coefficients

of a parity check code $\mathcal{C}$ so that $h_i = \alpha^{a_i}$ verifies $h_1 = \alpha^0$ (or $a_1 = 0$) and $i \leq j \Rightarrow a_i \leq a_j$. In the sequel, this convention will be used by default.

Since $\mathbf{X} \in \mathcal{C}$ is a vector of $GF(\mathrm{q})^{d_c}$, it is possible to determine its binary image to define a binary code of length $(md_c, m(d_c - 1))$. The Hamming weight spectrum $\mathcal{S}[X]$ of this code is defined as

$$\mathcal{S}[X] = 1 + S_1 X + S_2 X^2 + S_3 X^3 + \ldots + S_{md_c} X^{md_c}, \tag{3}$$

where S_n is the total number of codewords of Hamming weight n of the code. By convention, for a given set of coefficients H, $S_n(H)$ will denote the value of the n^{th} coefficient of the Hamming weight spectrum of the code defined by the set of coefficients H. The computation of the spectrum can be performed with a complexity of $q^2(d_c - 1) + 2q$ using the recursive algorithm used to compute the spectrum distance of a convolutional code [6]. The adaptation of the algorithm is given in Algo. 1. The partial spectrum $\mathcal{S}_y(l)[X]$, with $y \in$ GF(q), $l = 0, 1, 2, \ldots d_c$ represents the spectrum of codewords $(x_1, x_2, \ldots x_l)$ of size l that verify

$$\sum_{i=1}^{l} h_i x_i = y. \tag{4}$$

Note that when $l = 0$, we will assume that $\mathcal{S}_y(0)[X] = 1$ if $y = 0$ (empty set is a solution), 0 otherwise (there is no solution).

Moreover, it is possible to attribute also a Hamming Spectrum $\mathcal{S}^x[X]$ to an element of $x \in$ GF(q). It is the monomial $\mathcal{S}^x[X] = X^{W(x)}$ where $W(x)$ is the binary Hamming weight of x, i.e., the number of 1 in the polynomial representation of x.

In an Additive White Gaussian Noise (AWGN) channel, it is well known that the performance is determined first by the Hamming distance of the code (the index d_{min} of the smallest non null value of $\mathcal{S}[X]$, i.e. $S_{d_{min}} \neq 0$ while $0 < i < d_{min} \Rightarrow S_i = 0$) and second by the multiplicity of code word of minimum Hamming distance, i.e., the value $S_{d_{min}}$.

Theorem 1: Let H be a set of d_c coefficients in GF$(2^m)^{d_c}$, then

$$S_1(H) = 0. \tag{5}$$

Proof: Let us assume that there exits a vector $\mathbf{X}$ solution of (2) with a Hamming binary weight

Data: Initial set of coefficients H

Result: Spectrum $\mathcal{S}[X]$

$\mathcal{S}_s(0)[X] = 1$ if $s = 0$, 0 otherwise.

for $l = 1, \ldots d_c$ **do**

  for $d \in$ *GF(q)* **do**

    $\mathcal{S}_d(l)[X] = 0$

  end

  for $s \in$ *GF(q)* **do**

    for $x \in$ *GF(q)* **do**

      $d = s + (h_l x)$;

      $\mathcal{S}_d(l)[X]$ += $\mathcal{S}_s(l-1)[X]\mathcal{S}^x[X]$;

    end

  end

end

$\mathcal{S}[X] = \mathcal{S}_0(d_c)[X]$

Algorithm 1: Computation of Hamming weight spectrum associated to a parity check code

of 1. A binary hamming weight implies that there is only one non null value in vector $\mathbf{X}$. Let us assume that this value is x_i. Then (2) reduces to $x_i h_i = 0$. Since h_i is non null, then this implies that $x_i = 0$ which is in contradiction with the hypothesis ■

Theorem 2: Let H be a set of d_c coefficients in GF$(2^m)^{d_c}$, then

$$S_2(H) = \sum_{i=1}^{d_c-1} \sum_{j=i+1}^{d_c} S_2(\{h_i, h_j\}). \tag{6}$$

Proof: The demonstration of theorem 1 implies that a non-null vector $\mathbf{X}$ verifying (2) contains at least two non null GF values. If it contains more than two non-null values, then the binary Hamming weight will be greater than 2. Thus, binary Hamming weight two implies exactly two non null coefficients. The total number of codewords of binary Hamming weight two $S_2(H)$ is thus the summation of the number of codewords of binary Hamming weight two associated to each distinct couple of coefficients ■

Theorem 3: Let H be a set of d_c coefficients in GF$(2^m)^{d_c}$, then

$$S_3(H) = S_3^t(H) - (d_c - 3)S_3^c(H) \tag{7}$$

where the term $S_3^t(H)$ indicates the summation of binary Hamming weight 3 associated to all possible triplets of non-null coefficients, i.e.,

$$S_3^t(H) = \sum_{1 \leq i < j < k \leq d_c} S_3(\{h_i, h_j, h_k\}), \tag{8}$$

and the term $S_3^c(H)$ indicates the summation of binary Hamming weight 3 associated to all possible couples of non-null coefficients, i.e.,

$$S_3^c(H) = \sum_{1 \leq a < b \leq d_c} S_3(\{h_a, h_b\}), \tag{9}$$

Proof: Let us consider a triplet $\{h_i, h_j, h_k\}$ of coefficients of a parity check of degree 3. The set $\mathcal{C}$ of triplets (x_i, x_j, x_k) verifying $x_i h_i + x_j h_j + x_k h_k = 0$ can be partitioned in four different sets: $\mathcal{C}_i^{j,k} = (0, x_j, x_k)$, $\mathcal{C}_j^{i,k} = (x_i, 0, x_k)$, $\mathcal{C}_k^{i,j} = (x_i, x_j, 0)$ and $\mathcal{C}^{i,j,k} = (x_i, x_j, x_k)_{x_i \neq 0, x_j \neq 0, x_k \neq 0}$. Thus, $S_3(\{h_i, h_j, h_k\}) = S_3(\mathcal{C}_i^{j,k}) + S_3(\mathcal{C}_k^{i,k}) + S_3(\mathcal{C}_k^{i,j}) + S_3(\mathcal{C}^{i,j,k})$. One can note that $S_3(\mathcal{C}_k^{i,j})$ is independent of k and is equal to $S_3(\{h_i, h_j\})$. Moreover, in the computation of $S_3^t(H)$, a given couple (a, b), $a < b$ appears exactly in $d_c - 2$ triplets. Thus, $S_3(\{h_a, h_b\})$ contributes $(d_c - 2)$ times in the summation of $S_3^t(H)$. Thus $S_3^t(H) - (d_c - 2)S_3^c(H)$ gives the exact enumeration of weight 3 codewords with 3 non null values, while $S_3^c(H)$ gives the exact enumeration of weight 3 codewords with exactly two non-null terms. According to theorem 1, there is no solution with exactly one non-null term. Thus, adding those two terms gives the exact number of weight 3 codewords ■

Property 1 Let $x = \alpha^a$ an element of GF(2^m), then $W(x) = 1$ is equivalent to $0 \leq a < m$. In others words, the binary representation of x contains exactly one non null value (the binary Hamming weight of x is equal to 1, or $S^x[X] = X^1$)" is equivalent to the property $0 \leq a < m$. For example, if GF(2^3) is defined by $P_3[X] = 1 + X + X^3$, then $\alpha^0 = (1, 0, 0)$, $\alpha^1 = (0, 1, 0)$,

$\alpha^2 = (0,0,1)$ while $\alpha^3 = (1,1,0)$.

Theorem 4: Let $H = \{\alpha^{a_i}\}_{i=1,\ldots,d_c}$ be a set of d_c non null coefficients in GF$(2^m)^{d_c}$, then, if $m > 2$, $S_2(H) = 0$ is equivalent to

$$\forall i,j \in \{1,2,\ldots d_c\}^2, i \neq j \Rightarrow |a_j - a_i|_{q-1} \geq m, \tag{10}$$

where $|a|_{q-1}$ represents $\min(|a|, |q-1-a|)$.

Proof: Let us first prove the equivalence for a check node of degree $d_c = 2$ with the set of coefficients $\{h_1, h_2\}$, where $h_1 = \alpha^{a_1}$ and $h_2 = \alpha^{a_2}$ and $a_2 \geq a_1$. Multiplying the coefficients of the check node by α^{-a_1} does not change the code, thus, h_1 can be set to $h_1 = \alpha^0$ and h_2 can be set to α^a; with $a = a_2 - a_1$. The $q-1$ non null solutions of the parity check equation are thus $(x_1^b = \alpha^{b+a}, x_2^b = \alpha^b)$, $b = 0,1,\ldots q-2$. In fact, $h_1 x_1^b + h_2 x_2^b = \alpha^0\alpha^{b+a} + \alpha^a\alpha^b = \alpha^{a+b} + \alpha^{a+b} = 0$. For a given b, the Hamming weight of the codeword (x_1^b, x_2^b) is equal to $W(x_1^b) + W(x_2^b)$. According to property 1, We have $W(x_1^b) = 1$ equivalent to $0 \leq a + b \mod q - 1 < m$ or equivalently

$$(0 \leq b < m - a) \text{ or } (q - 1 - a \leq b < q - 1). \tag{11}$$

Similarly, $W(x_2^b) = 1$ is equivalent to

$$0 \leq b < m. \tag{12}$$

Thus, according to theorem 1, $W(x_1^b) + W(x_2^b) = 2 \Rightarrow W(x_1^b) = 1$ and $W(x_2^b) = 1$, or equivalently, there exists a value of b that satisfies simultaneously (11) and (12). There is a solution if and only if $0 \leq m - 1 - a$ or $q - 1 - a \leq m - 1$. If $m > 2$, the second inequality is never fulfilled and the existence of solution is given by $a \leq m - 1$. Reciprocally, for $m > 2$, if $a > m - 1$, then $W(x_1^b) + W(x_2^b)$ is always strictly greater than 2. The general case can be proven by using theorem 2 ■

Theorem 5: Let us consider a parity check of degree d_c over GF(2^m). Then, there exits $H \in GF(2^m)^{d_c}$ so that $S_2(H) = 0$ is equivalent to $d_c \leq \frac{2^m}{m}$.

Proof: $H = \{\alpha^0, \alpha^m, \alpha^{2m}, ..., \alpha^{(d_c-1)m}\}$ verifies (10) if and only if $(d_c - 1)m \leq q - m$ ■

Our objective in this paper is thus to find, for several values of d_c and Galois Field GF(2^m) the sets of coefficients that minimize $S_3(H)$ with $S_2(H) = 0$. The design objective can be formalized as

$$H_3^{opt} = \arg \min_{H \in GF(2^m)^{d_c}} \{S_3(H)/S_2(H) = 0\}. \tag{13}$$

The next sections show the method used to reach this objective.

III. Estimation of the complexity

It is useful to compute the exact number of configurations to be tested in order to explore all possible sets of coefficients leading to $S_2(H) = 0$. To do so, we use a method inspired from the Pascal' Triangle method [7].

Let $\Gamma_m(p, n)$ be the set of p-uplet of integer $(a(1), a(2), \ldots, a(p))$ verifying the following two constraints

$$a(i) \in \{0, 1, \ldots n-1\},\ i = 1, 2, \ldots, p \tag{14}$$

and

$$a(i+1) - a(i) \geq m,\ i = 1, 2, \ldots, p-1 \tag{15}$$

The cardinality $|\Gamma|$ of set Γ will be denoted as $\gamma = |\Gamma|$. According to this definition, $\Gamma_6(2, 8)$ is equal to $\Gamma_6(2, 8) = \{(0, 6), (0, 7), (1, 7)\}$ and the cardinality of $\Gamma_6(2, 8)$ is $\gamma_6(2, 8) = 3$.

Case $p = 1$: When $p = 1$, then only constraint (14) can be applied and thus $\Gamma_m(1, n) = \{0, 1, \ldots, n\}$ and $\gamma_m(1, n) = n$.

Case $p = 2$: When $p = 2$, if $n \leq m$, there is no solution, thus $\Gamma_m(2, n) = \emptyset$ and $\gamma_m(2, n) = 0$.
If $n = m + 1$, there is a unique solution $\Gamma_m(2, m + 1) = \{(0, m)\}$ and thus $\gamma_m(2, m + 1) = 1$.
If $n = m + 2$, there are 3 possible solutions: $\Gamma_m(2, m + 1) = \{(0, m), (0, m + 1), (1, m + 1)\}$ and $\gamma_m(2, m + 1) = 3$.
If $n = m+3$, there are 6 elements $\Gamma_m(2, m+3)$. In fact, the elements of $\Gamma_m(2, m+2)$ belongs also

to $\Gamma_m(2, m+3)$. The additional elements are the 3 couples $(0, m+2)$, $(1, m+2)$ and $(2, m+2)$. These 3 couples can be represented by $\{\Gamma_m(1, n-m) \parallel m+2\}$, where $\{\Gamma||x\}$ means the set obtained by concatenating x on the right to all elements of Γ. In other words, $\Gamma_m(2, m+3) = \Gamma_m(2, m+2) \cup \{\Gamma_m(1, n-m)||m+2\}$, and thus, $\gamma_m(2, m+3) = \gamma_m(2, m+2) + \gamma_m(1, n-m)$.

General Case: In the general case, $\Gamma_m(2, n) = \Gamma_m(2, n-1) \cup \{\Gamma_m(1, n-m)||n-1\}$ and thus

$$\gamma_m(2, n) = \gamma_m(2, n-1) + \gamma_m(1, n-m). \tag{16}$$

In (16), we recognize the structure of the Pascal's triangle binomial construction, and thus

$$\gamma_m(2, n) = \binom{2}{n-m+1} = \frac{(n-m+1)(n-m)}{2}. \tag{17}$$

In the general case, $\Gamma_m(p, n)$ and $\gamma_m(p, n)$ can be determined by a double recursive equation. First, let us assume that $\Gamma_m(p', n')$ are known for all couples $(p' < p, n' \in \mathbb{N})$ and $(p, n' < n)$. Then, $\Gamma_m(p, n)$ can be generated as

$$\Gamma_m(p, n) = \Gamma_m(p, n-1) \cup \{\Gamma_m(p-1, n-m)||n-1\}. \tag{18}$$

This equality gives

$$\gamma_m(p, n) = \gamma_m(p, n-1) + \gamma_m(p-1, n-m). \tag{19}$$

The derivation of the exact value of $\gamma_m(p, n)$ is out of the scope of this paper. We can nevertheless derive from the recursion method that

$$\gamma_m(p, n) = \sum_{k=1}^{n} \gamma_m(p-1, k-m). \tag{20}$$

The important point is that the exact number of configurations can be, in practice, determined. As an example, Tab. I gives the first values of $\gamma_5(p, n)$ for $p \leq 5$ and $n \leq 22$

Let us go back to our initial problem: determine $\xi_m(d_c)$, the number of sets of coefficients in a check node of degree d_c over GF(2^m) that gives a minimum Hamming weight 3 for its equivalent binary code. The first coefficient can be always $h_1 = \alpha^0$, since the multiplication of all coefficients by the same constant value doesn't change the code. Once α^0 is selected, $\{\alpha^1, \alpha^2, \ldots, \alpha^{m-1}\}$ and $\{\alpha^{q-m-1}, \ldots, \alpha^{q-3}, \alpha^{q-2}\}$ are removed in order to respect theorem 4, i.e., every pair of coefficients of the check node should have their logarithms separated by at

value of n	$n \leq 5$	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
$p=1$	n	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
$p=2$	0	1	3	6	10	15	21	28	36	45	55	66	78	91	105	120	136	153
$p=3$	0	0	0	0	0	0	1	4	10	20	35	56	84	120	165	220	286	364
$p=4$	0	0	0	0	0	0	0	0	0	0	0	1	5	15	35	70	126	210
$p=5$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	6

TABLE I

VALUES OF $\gamma_5(p,n)$

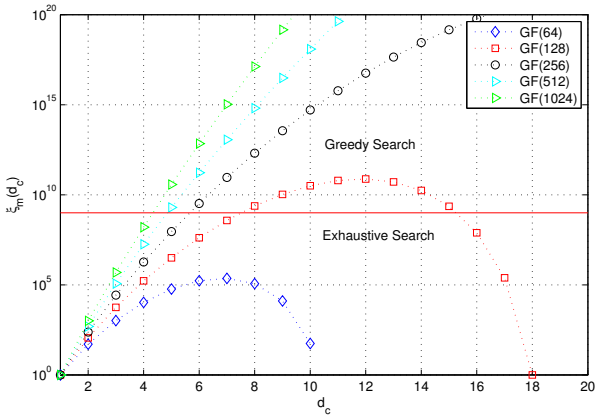

Fig. 1. Number $\xi(d_c, m)$ of sets of coefficients for $d_c \leq 20$ over GF(64) up to GF(1024)

least m. Thus, there are still $p = d_c - 1$ points to be placed among $2^m - 1 - (2m-1) = 2^m - 2m$ values (see Fig. 2.a)

$$\xi_m(d_c) = \gamma_m(d_c - 1, 2^m - 2m). \tag{21}$$

From (21) and Table I, we deduce that there is exactly $\xi_5(4) = \gamma_5(3, 22) = 364$, i.e., there is 364 sets of coefficients, with the first one equal to α^0, that lead to a Hamming distance of 3 for a check node of degree 4 over GF($2^m = 32$) (coefficients are supposed to be sorted in increasing order of their logarithm). It is thus easy to generate these 364 solutions in order to keep the ones leading to the minimum multiplicity of weight 3 codewords (i.e., the minimum of $S_3(H)$). Fig. 1 shows the number of configurations $\xi_m(d_c)$ for m equal to 6 (GF(64)) to 10 (GF(1024)) and d_c varying from 3 to 20.

Note that $\xi_8(20) = 2.39 \times 10^{22}$ (not shown in Fig. 1), which is a far too high number for an exhaustive search. In this paper, we limit the exhaustive exploration to solution where

$\xi_m(d_c) < 10^9$ in order to get the optimal solution. When $\xi_m(d_c) \geq 10^9$, a heuristic search should be used to find good sets of coefficients.

IV. Heuristic search of coefficients

When the value of $\xi_m(d_c)$ is too high for an exhaustive exploration, a heuristic search should be used. In this paper, we propose a basic but effective method. It is based on a greedy search repeated several times, each attempt starting from an initial state taken randomly. Let N_g be the number of attempt, $H^{0,k}$ the k^{th} random initial set of coefficients, $G(H^{0,k})$ the final state obtained when a greedy algorithm is applied on $H^{0,k}$. The final solution H_3^f is taken as

$$H_3^f = \arg\min\{S_3(G(H^{0,k})), k = 1 \dots N_g\}. \tag{22}$$

Let us describe in more details the method to draw the $H^{0,k}$ and the greedy algorithm.

A. Method to generate initial sets of coefficients

The generation of the initial set should be unbiased, i.e., any element of $\xi_m(d_c)$ should have the same probability $P = \frac{1}{|\xi_m|}$ of being chosen. This requirement can be achieved by a step by step generation process. In the sequel, the index k is omitted for clarity.

The first element h_1^0 of H^0 is always $h_1^0 = \alpha^0$. Then, the smallest (in the sense of logarithm over GF) next element is $h_2^0 = \alpha^m$ ($a_2 = m$) according to theorem 4. In that case, there are still $d_c - 2$ coefficients to be drawn among $2^m - 3m$ positions, as shown if Fig. 2.b. The number of elements is thus $\gamma_m(d_c - 2, 2^m - 3m)$ possibilities. If the next chosen element is $a_2 > m$, as shown in 2.c, there is still $d_c - 2$ coefficients to be drawn among $2^m - a_2 - 2m$, and thus $\gamma_m(d_c - 2, 2^m - a_2 - 2m)$ possibilities. In order to draw a set of coefficients randomly, we should have, for the second coefficient:

$$\text{Prob}(h_2^0 = \alpha^{a_2}) = \frac{\gamma_m(d_c - 2, 2^m - a_2 - 2m)}{\gamma_m(d_c - 1, 2^m - 2m)}. \tag{23}$$

One should note that the sum of the probability $\text{Prob}(h_2^0 = \alpha^{a_2})$ for all values of a_2 is equal to 1 according to (20). For the third element (and the fourth up to the last one), the same method can be applied, leading to the general formula to generate the j^{th} coefficients a_j knowing that the previous coefficient is a_{j-1}, $a_j > a_{j-1}$ is given by

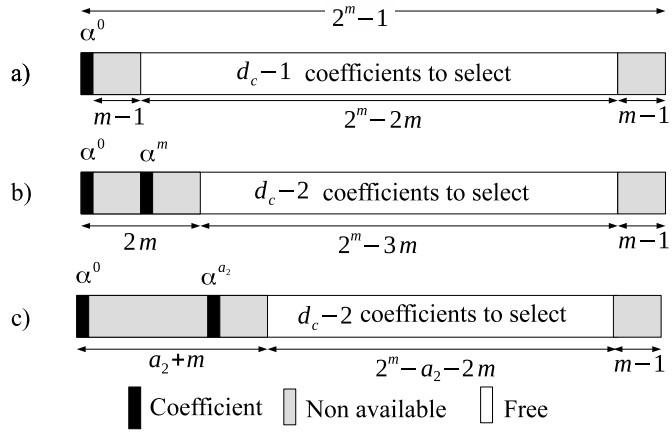

Fig. 2. Illustration of the random coefficients selection process.

$$\text{Prob}(h_j^0 = \alpha^{a_j} / h_{j-1}^0 = \alpha^{a_{j-1}}) = \frac{\gamma_m(d_c - j, 2^m - 2m - a_j)}{\gamma_m(d_c - j + 1, 2^m - 2m - a_{j-1})}. \tag{24}$$

To conclude, the generation of uniformly distributed sets of coefficients reduces to a Markov process where probability of transition at a given stage is given by (24).

B. Proposed greedy algorithm

The initial set of H coefficient is $H = H^0$, then all possible values for the second coefficient h_2 verifying $h_2 = \alpha^{a(2)}$, with $m \leq a(2) \leq a(3) - m$ are tested. This limited search space guaranties that $S_2(H) = 0$ (see Theorem 4). The value of $a(2)$ that minimizes S_3 is selected to generate the new set of coefficients H. Then the same process is applied on the third coefficient (with $a(2) + m \leq a(3) \leq a(4) - m$) up to the d_c^{th} coefficient. The whole process is started again until no more improvement is obtained. The algorithm is given in details in algorithm 2. Note that when $l = d_c$, $l+1$ goes back to 1, and thus, $a_{d_c+1} - m = -m \mod 2^m - 1 = 2^m - m$. One should note that many more sophisticated and efficient algorithms can be imagined. Nevertheless, repeated many times from random initial states, the overall search method is effective.

Fig. 3 shows the histogram of $S_3(H^0)$ obtained with $N = 20,000$ draws as well as the best value found for $d_c = 6, 8, 10$ and 12 over GF(256). In order to evaluate how far is the best found solution S_3^f compared to the average value of $S_3(H^0)$, we use the two following metrics

Data: Initial set of coefficients $H^0 = \{\alpha^{a(i)}\}_{i=1,...d_c}$

Result: Final set of coefficients H_3^f

Improved = true;

$s_{opt} = S_3(H)$;

$G(H^0) = H^0$

while *Improved* **do**

  Improved = false;

  for $l = 2, \dots d_c$ **do**

    $H = G(H^0)$;

    for $b = a(l-1) + m$, ..., $a(l+1) - m$ **do**

      $h_l = \alpha^b$;

      $s = S_3(H)$;

      if $s < s_{opt}$ **then**

        $G(H^0)(l) = \alpha^b$;

        $s_{opt} = s$;

        Improved = true;

      end

    end

  end

end

Algorithm 2: Greedy algorithm to compute $G(H^0)$

$$\Delta_3 = \frac{M_3 - S_3^f}{\sigma_3} \tag{25}$$

$$R_3 = \frac{S_3^f}{M_3} \times 100 \ \ (\text{in } \%) \tag{26}$$

where M_3 and σ_3 are respectively the mean and the standard deviation of $S_3(H^0)$ for H^0 satisfying $S_2(H^0) = 0$. The first metric Δ_3 measures how far is the found value relatively to the "gaussian like shape" distribution of $S_3(H^0)$ while the second metric indicates the relative gain, in %, compared to the mean value M_3. Fig. 4 and Fig. 5 show the evolution of Δ_3 and M_3 for several values of d_c and GF order. One can note that the curves for GF(128) and GF(256) show smooth variations while curves for GF(512) and GF(1024) show some irregularities. A probable

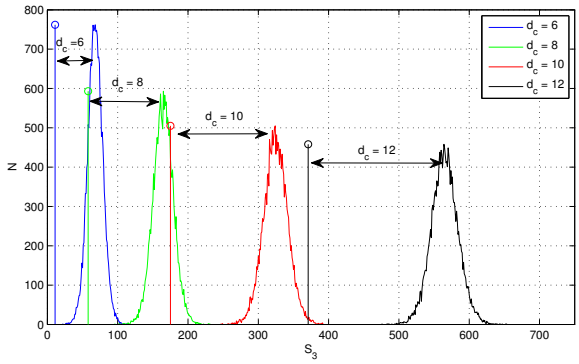

Fig. 3. Histogram of $S_3(H^0)$ and the best value S_3^{opt} over GF(256) for several values of check node degree d_c

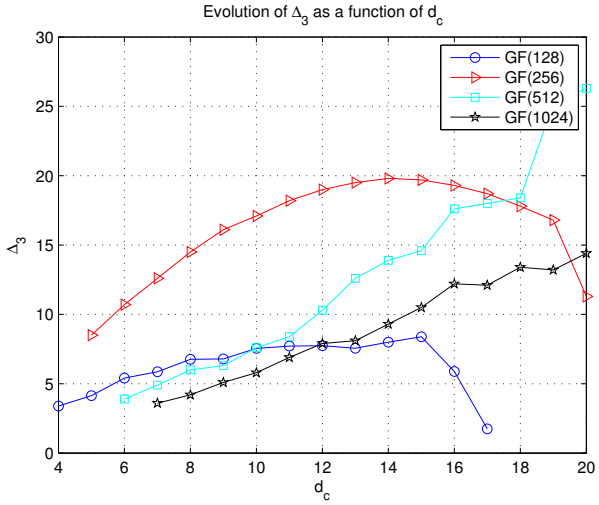

Fig. 4. Value of Δ_3 as a function of d_c for GF(64) up to GF(1024)

explanation is the inefficiency of the greedy algorithm that requires a lot of time per trial for those high order Galois Field. In fact, H_3^f are obtained with 20000 trials up to GF(256), but with only 100 trials for GF(1024). In other words, there are probably sets of coefficients that lead to slightly smallest S_3^f values for those high order fields. The search for better set of coefficients for GF(512) and GF(1024) is still open.

Values of H_3^f, M_3, σ_3 and the corresponding set of coefficients are given for GF(64) up to GF(1024) in annex 1.

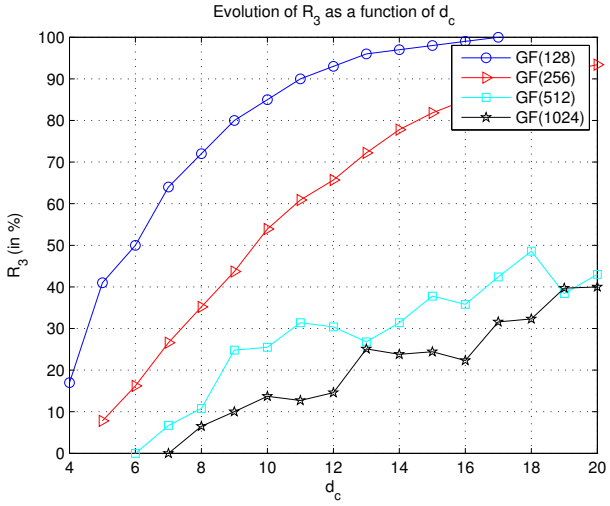

Fig. 5. Value of M_3 as a function of d_c for GF(64) up to GF(1024)

V. Conclusion

In this paper, we have generalized the method proposed by Pouillat et al. for the determining the optimal Galois Field coefficients of a Non-Binary LDPC parity check code based on the binary image of the code. An algorithm with a complexity in $(O)(d_c^2)$ has been proposed to determine the number $S_3(H)$ of codewords of binary Hamming weight 3 of a parity check of degree d_c over GF(q). The low computational complexity of the algorithm opens exploration to new regions of the design space, i.e. check node degree d_c greater than 4 and high order Galois Field (up to GF(1024)) by an exhaustive search. A new greedy search algorithm has also been proposed to find good solutions when the number of sets of coefficients is too high for an exhaustive search. Tables of sets of coefficients are given for values of d_c between 4 and 20 and GF order varying from 64 to 1024. For each set of coefficients, the best found value $S_3^f(H)$ is compared with the distribution of $S_3(H)$ obtained by taking randomly the coefficients of H. In some cases, $S_3^f(H)$ can be at a distance to the mean value of $S_3(H)$ greater than 10 times the standard deviation of the distribution. The proposed sets of coefficients can effectively replace the random selection of coefficients often used in NB-LDPC construction over high order Galois Field, and thus helps the construction of new generations of NB-LDPC codes with better decoding performance.

ACKNOWLEDGEMENT

The author would like to thank Cédric Marchand that point out some errors in the preliminary of the paper.

VI. ANNEX

In the annex, we give the results obtained by the proposed methods in order to help the construct optimal, or almost optimal, NB parity check codes. Note that multiplying a set of coefficients does not change the code. For example $H = \{\alpha^1, \alpha^{10}, \alpha^{23}, \alpha^{38}\}$ over GF(64) gives the same code as $H' = H\alpha^{53} = \{\alpha^{53}, \alpha^{63}, \alpha^{76}, \alpha^{91}\} = \{\alpha^{53}, \alpha^1, \alpha^{13}, \alpha^{28}\}$. After reordering of the coefficients, H' is equal to $H' = \{\alpha^1, \alpha^{13}, \alpha^{28}, \alpha^{53}\}$. Since the parity check generated by H, $H' = H\alpha^{53}$, $H'' = H\alpha^{40}$ and $H''' = H\alpha^{25}$ are all equal, only the set of coefficients that minimizes the value of a_2 will be given to represent the equivalent set of coefficients through a multiplicative factor. When distinct optimal solutions exist for a given configuration of d_c and GF(q), those solutions are enumerated.

REFERENCES

[1] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, pp. 1645–, Aug 1996.

[2] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular (2,d/sub c/)-ldpc codes over gf(q) using their binary images," *IEEE Transactions on Communications*, vol. 56, no. 10, pp. 1626–1635, October 2008.

[3] D. Mackay. Optimizing sparse graph codes over gf(q). [Online]. Available: http://www.inference.org.uk/mackay/CodesGallager.html

[4] L. Dolecek, D. Divsalar, Y. Sun, and B. Amiri, "Non-binary protograph-based ldpc codes: Enumerators, analysis, and designs," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 3913–3941, July 2014.

[5] A. E. Brouwer and T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes," *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 662–677, Mar 1993.

[6] M. Rouanne and D. J. Costello, "An algorithm for computing the distance spectrum of trellis codes," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 6, pp. 929–940, Aug 1989.

[7] B. Pascal, *Traité du triangle arithmétique avec quelques autrez petits traitez sur la même matière*, 1654.

[8] C. Poulliat, M. Fossorier, and D. Declercq, "Using binary images of non binary ldpc codes to improve overall performance," in *4th International Symposium on Turbo Codes Related Topics; 6th International ITG-Conference on Source and Channel Coding*, April 2006, pp. 1–6.

d_c	S_3^f	M_3	σ_3	Δ_3	R_3 (%)	GF(64)
4^a	^{*20}	31.5	3.3	3.5	63.4 %	$\{0,9,22,37\}$
5^b	^{*51}	65.0	3.5	4.0	78.4 %	$\{0,6,17,43,52\}$; $\{0,6,32,41,54\}$; $\{0,7,17,43,52\}$; $\{0,7,18,43,52\}$
						$\{0,7,18,44,53\}$; $\{0,7,18,44,52\}$; $\{0,7,32,41,54\}$; $\{0,7,33,41,54\}$
						$\{0,7,33,42,54\}$; $\{0,7,33,42,55\}$
6	^{*100}	115.9	3.6	4.4	86.3 %	$\{0,6,12,19,45,54\}$; $\{0,6,13,19,45,54\}$; $\{0,6,13,20,45,54\}$
						$\{0,6,13,20,46,55\}$
7	^{*173}	187.9	3.1	4.8	92.1 %	$\{0,6,12,20,27,43,53\}$; $\{0,6,13,21,27,43,53\}$; $\{0,6,13,21,28,43,53\}$
						$\{0,6,13,21,28,44,53\}$; $\{0,6,13,21,28,44,54\}$
8	^{*276}	283.3	1.7	4.0	97.4 %	$\{0,6,12,20,27,35,43,53\}$; $\{0,6,13,20,27,35,43,53\}$
						$\{0,6,13,21,27,35,43,53\}$; $\{0,6,13,21,28,35,43,53\}$
						$\{0,6,13,21,28,36,43,53\}$; $\{0,6,13,21,28,36,44,53\}$
						$\{0,6,13,21,28,36,44,54\}$
9	^{*402}	406.8	1.0	5.0	98.8 %	$\{0,6,14,21,27,35,42,48,55\}$; $\{0,6,14,21,27,35,42,48,56\}$
10	^{*560}	560.9	0.2	4.1	99.8 %	$\{0,6,12,18,24,30,37,44,50,56\}$; $\{0,6,12,19,25,31,37,43,49,55\}$
						$\{0,6,12,19,25,31,37,43,49,56\}$

[a]This set of coefficients was initially proposed in [3] and [8]

[b]In [3], a list of 77 of sets of coefficients are given for $d_c = 5$ over GF(64). In this list, some sets of coefficients have $S_2(H) > 0$. The best proposed one is $H = \{\alpha^1, \alpha^7, \alpha^{36}, \alpha^{58}\}$ with $S_2(H) = 0$ and $S_3(H) = 57$.

TABLE II

LIST OF OPTIMAL COEFFICIENT'S EXPONENTS $\{a_i\}_{i=1,\dots d_c}$ FOR GF(64). THE SYMBOL * INDICATES THAT THE VALUE OF S_3^f IS EQUAL TO S_3^{opt}.

d_c	S_3^f	M_3	σ_3	Δ_3	R_3 (%)	GF(128),
4[a]	*4	23.3	5.7	3.39	17 %	$\{0, 11, 84, 101\}$; $\{0, 12, 84, 101\}$
5	*20	49.0	7.0	4.14	41 %	$\{0, 10, 21, 94, 111\}$; $\{0, 11, 55, 84, 101\}$
6	*44	87.8	8.1	5.41	50 %	$\{0, 9, 21, 60, 93, 112\}$; $\{0, 9, 21, 60, 94, 112\}$; $\{0, 9, 21, 61, 93, 112\}$
7	*92	143.0	8.7	5.86	64 %	$\{0, 7, 24, 39, 48, 60, 99\}$
8	*157	217.2	8.9	6.76	72 %	$\{0, 7, 16, 33, 50, 59, 71, 111\}$
9	*252	313.1	9.0	6.79	80 %	$\{07, 19, 30, 37, 53, 68, 77, 89\}$
10	370	433.4	8.4	7.55	85 %	$\{0, 7, 38, 45, 59, 68, 75, 91, 106, 115\}$
11	522	581.4	7.7	7.71	90 %	$\{0, 7, 22, 30, 37, 48, 55, 69, 78, 89, 96\}$
12	709	759.3	6.5	7.74	93 %	$\{0, 7, 18, 25, 39, 48, 59, 66, 88, 97, 104, 118\}$
13	928	969.6	5.5	7.56	96 %	$\{07, 17, 24, 38, 48, 58, 65, 72, 87, 96, 103, 117\}$
14	1182	1215.6	4.2	8.00	97 %	$\{0, 7, 14, 29, 38, 45, 55, 62, 69, 76, 86, 93, 106, 115\}$
15	1473	1499.0	3.1	8.39	98 %	$\{0, 7, 18, 25, 32, 39, 47, 54, 61, 69, 78, 89, 96, 103, 118\}$
16	*1813	1823.0	1.7	5.88	99 %	$\{0, 7, 17, 24, 31, 38, 47, 54, 61, 68, 75, 82, 89, 96, 103117\}$
17	*2190	2190.7	0.4	1.75	100 %	$\{0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98, 105, 118\}$
18	*2604	2604.0	0.0	0.0	100 %	{0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98, 105, 112, 119}

[a]In [8], the best given sets of coefficients have $S_3(H) = 5$

TABLE III

LIST OF OPTIMAL OR OPTIMIZED (N_g = 20,000) SETS OF COEFFICIENT'S EXPONENTS $\{a_i\}_{i=1,...d_c}$ FOR GF(128). THE SYMBOL * INDICATES THAT THE VALUE OF S_3^f IS EQUAL TO S_3^{opt}.

d_c	S_3^f	M_3	σ_3	Δ_3	R_3 (%)	GF(256)
4[a]	*0	19.2	6.3	3.0	0 %	$\{0,8,172,183\}$; $\{0,8,172,182\}$; $\{0,8,171,182\}$
5	*3	38.6	4.2	8.5	7.8 %	$\{0,8,66,172,180\}$
6	11	68.1	5.3	10.7	16.2 %	$\{0,8,75,83,91,149\}$
7	29	109.2	6.3	12.6	26.6 %	$\{0,8,76,84,92,131,150\}$
8	58	164.5	7.3	14.5	35.2 %	$\{0,8,36,75,83,91,128,148\}$
9	103	235.6	8.2	16.1	43.7 %	$\{0,8,37,76,84,92,129,149,233\}$
10	175	324.7	8.8	17.1	53.9 %	$\{0,8,16,54,74,139,158,178,187,214\}$
11	264	433.6	9.3	18.2	60.9 %	$\{0,8,27,92,109,131,139,169,208,216,224\}$
12	371	564.9	10.2	19.0	65.7 %	$\{0,8,27,39,92,109,132,140,169,208,216,224\}$
13	522	720.0	10.2	19.5	72.2 %	$\{0,8,18,38,46,65,77,130,147,170,178,207,245\}$
14	701	901.3	10.1	19.8	77.8 %	$\{0,8,16,42,82,90,98,107,128,136,154,166,219,236\}$
15	908	1110.3	10.3	19.7	81.8 %	$\{0,8,29,37,76,84,92,103,123,131,150,162,192,214,232\}$
16	1150	1349.8	10.6	19.3	84.9 %	$\{0,8,16,34,42,79,87,95,106,126,134,153,165,195,217,235\}$
17	1426	1621.3	10.5	18.7	88.0 %	$\{0,8,45,53,61,69,77,94,102,121,133,164,186,203,221,229,237\}$
18	1737	1926.9	10.7	17.8	90.1 %	$\{0,8,19,27,35,52,60,92,100,108,116,126,147,155,173,185,216,237\}$
19	2083	2268.5	11.0	16.8	91.8 %	$\{0,8,26,39,70,91,109,117,126,134,142,161,169,183,202,210,218,226,236\}$
20	2473	2648.4	15.1	11.3	93.4 %	$\{0,8,22,30,38,52,61,75,93,101,109,117,127,147,155,174,186,206,216,238\}$

[a]The set $H = \{0,8,172,183\}$ is also given in [8]. Note that for these 3 sets, $S_4(H)$ is minimal and equal to 156.

TABLE IV

LIST OF OPTIMAL OR OPTIMIZED (N_g = 20,000) SETS OF COEFFICIENT'S EXPONENTS $\{a_i\}_{i=1,\ldots d_c}$ FOR GF(256). THE SYMBOL * INDICATES THAT THE VALUE OF S_3^f IS EQUAL TO S_3^{opt}.

d_c	S_3^f	M_3	σ_3	Δ_3	R_3 (%)	Optimized ($N_g = 1,000$) coefficient's exponents $\{a_i\}_{i=1,\ldots d_c}$ for GF(512)
6	0	45.3	11.6	3.9	0 %	$\{0, 20, 120, 157, 390, 474\}$
7	5	73.8	14.2	4.9	6.7 %	$\{0, 20, 74, 159, 228, 312, 366\}$
8	12	111.8	16.7	6.0	10.8 %	$\{0, 20, 74, 119, 159, 228, 312, 366\}$
9	40	161.1	19.3	6.3	24.8 %	$\{0, 14, 49, 213, 288, 332, 353, 411, 441\}$
10	57	223.6	21.9	7.6	25.5 %	$\{0, 14, 64, 213, 232, 332, 354, 372, 441, 476\}$
11	94	299.5	24.3	8.4	31.4 %	$\{0, 14, 173, 212, 231, 287, 331, 352, 371, 410, 440\}$
12	119	391.7	26.6	10.3	30.4 %	$\{0, 14, 62, 212, 231, 287, 331, 353, 372, 410, 441, 477\}$
13	197	500.4	29.0	12.6	26.8 %	$\{0, 13, 120, 160, 180, 238, 281, 303, 322, 390, 427, 459, 474\}$
14	293	627.4	31.0	13.9	31.4 %	$\{0, 9, 35, 58, 115, 158, 180, 199, 237, 268, 304, 337, 352, 401\}$
15	338	775.3	33.1	14.6	37.8 %	$\{0, 12, 34, 53, 91, 122, 158, 179, 191, 205, 255, 365, 404, 424, 478\}$
16	481	943.4	34.4	17.6	35.8 %	$\{0, 12, 27, 76, 113, 192, 224, 243, 299, 343, 364, 384, 422, 437, 453, 490\}$
17	611	1135.2	36.4	18.0	42.4 %	$\{0, 11, 40, 59, 117, 148, 159, 180, 202, 253, 268, 305, 325, 337, 352, 402, 441\}$
18	800	1350.6	37.7	18.4	48.6 %	$\{0, 12, 34, 53, 91, 106, 121, 159, 179, 191, 205, 261, 294, 346, 363, 403, 422, 479\}$
19	981	1592.6	38.9	25.2	38.4 %	$\{0, 10, 40, 76, 97, 110, 124, 137, 161, 174, 207, 282, 322, 341, 397, 428, 441, 462, 484\}$
20	1217	1891.2	40.4	26.3	43.0 %	$\{0, 9, 42, 91, 108, 148, 167, 225, 244, 256, 268, 291, 310, 346, 377, 414, 434, 446, 461, 497\}$

TABLE V

LIST OF OPTIMIZED (N_g = 1000) SETS OF COEFFICIENT'S EXPONENTS $\{a_i\}_{i=1,\ldots d_c}$ FOR GF(512).

d_c	S_3^f	M_3	σ_3	Δ_3	R_3 (%)	GF(1024)
7	0	50.6	13.9	3.6	0.0 %	$\{0, 66, 207, 591, 684, 828, 955\}$
8	5	76.7	16.9	4.2	6.5 %	$\{0, 22, 61, 287, 478, 691, 826, 878\}$
9	11	110.4	19.6	5.1	10.0 %	$\{0, 23, 128, 241, 353, 471, 497, 666, 696\}$
10	21	153.3	22.7	5.8	13.7 %	$\{0, 22, 249, 410, 666, 730, 845, 901, 939, 986\}$
11	26	205.0	25.9	6.9	12.7 %	$\{0, 29, 163, 199, 229, 450, 554, 649, 808, 847, 991\}$
12	39	267.5	28.8	7.9	14.6 %	$\{0, 24, 52, 219, 258, 321, 452, 577, 618, 793, 818, 955\}$
13	68	342.4	31.7	8.7	19.9 %	$\{0, 16, 135, 181, 358, 382, 519, 586, 610, 638, 724, 845, 908\}$
14	102	428.6	35.2	9.3	23.8 %	$\{0, 25, 144, 259, 322, 421, 518, 546, 618, 675, 810, 859, 953, 990\}$
15	129	528.8	38.1	10.5	24.4 %	$\{0, 24, 52, 144, 219, 258, 322, 420, 575, 618, 675, 794, 822, 886, 955\}$
16	144	645.1	41.0	12.2	22.3 %	$\{0, 24, 52, 133, 219, 258, 321, 420, 575, 618, 675, 761, 791, 817, 953, 988\}$
17	245	775.9	43.9	12.1	31.6 %	$\{0, 22, 61, 125, 255, 378, 421, 478, 597, 622, 712, 759, 793, 825, 849, 877, 958\}$
18	298	922.5	46.7	13.4	32.3 %	$\{0, 14, 98, 126, 155, 198, 255, 341, 373, 535, 570, 602, 626, 735, 821, 860, 900, 923\}$
19	432	1087.2	49.8	13.2	39.7 %	$\{0, 21, 60, 179, 204, 243, 271, 339, 374, 407, 432, 460, 605, 666, 730, 771, 842, 938, 981\}$
20	507	1270.0	52.8	14.4	40.0 %	$\{0, 11, 59, 85, 124, 188, 235, 287, 300, 384, 412, 441, 484, 541, 660, 684, 820, 857, 889, 913\}$

TABLE VI

LIST OF OPTIMIZED (N_g = 100) SETS OF COEFFICIENT'S EXPONENTS $\{a_i\}_{i=1,...d_c}$ FOR GF(1024).