ADV MATH SCI JOURNAL
Advances in Mathematics: Scientific Journal **9** (2020), no.12, 11109–11125
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.12.91

APPLICATION OF A SPECIAL POLYNOMIAL TO AN ENTROPICOGENETIC CODING

H. S. G. RAVELONIRINA[1], J. J. RAKOTO, AND H. M. RAZAKASOA

ABSTRACT. The main purpose of this paper is to propose an another type of method of entropy coding. We will use a special polynomial and the genetic code. This is a prefix code with a variable length to basis 4. However, in order to verify the effectiveness of the code, the conformity to Shannon theorems, to Kraft inequality and to the calculation of the entropy of the source, we convert this code in binary with a view to stocking the information on computer. This coding is authentic: without error nor loss of symbol, including the white space. We have created an high-performance encoding and decoding software.

1. INTRODUCTION

Since time immemorial, the coding of information has always had a great importance in the life of man: of the learning of signs, of word then the writing. The morse code have been the first coding used for the communication with a length distance or hidden. It was Samuel F. B. Morse who developed it in 1844 [2]. This code is composed of points and dash (binary code). After the morse code, many codes have been invented such as:- The Baudot (or Murray) code invented by Emile Baudot in 1874 [2]- the phone developed by Dr Graham Bell on march 10, 1876 [21] who used the Baudot code- The ASCII code (American Standard Code for Information Interchange) which is adopted as standard was invented by Bob Bemer in 1961 [23](character coding on 8 bits). Despite the fact that this code is

[1]*corresponding author*

2020 *Mathematics Subject Classification.* 17B66, 17B56, 17B40, 17B70.

Key words and phrases. Alphabet, Messages, Special polynomial, Entropicogenetic coding.

standard for the character codings, there are also many others like the EBCDIC code (Extended Binary Coded Decimal Interchange Code) to 8 bits developed by IBM; the unicode code on 16 bits developed in 1991 [2]. At the end of the secund world war, in 1945, Claude Elwood Shannon discovered the first concepts of the coding theory published at the founder article. C. Shannon published in 1948 with W. Weaver an article "Mathematical theory of communications" which founded the basis of the code and information theory (cf. [21]). From this code David A. Huffman developed another code ensuring the optimality. The Shannon-Fano and Huffman codes have some common points. They are two codes with variable length, prefix and allow us to do a data compression and need the prior knowledge of the symbol probabilities. They belong to the same category of statistic coding [21]. The difference however is that the Huffman code is an optimal code but that of Shannon isn't. Both of them have limits, namely the probability of symbols can likely be unknown, and it can change over the time and the source might not generate symbols i.i.d (independent and identically distributed), for example the English text. Can the C. Shannon- Fano code and that of Huffman be used if we don't have a prior knowledge of the symbol probabilities of the source? The negative answer to this question allows us to propose an another type of the coding method: the coding that we call "Entropicogenetic encoding". We will see in this paper how to proceed to this type of coding. The principle basis of this coding is to replace the emitted messages by a source S using an alphabet $N-$area $\{s_1,\ s_2, \ldots,\ s_N\}$ by writing messages in the alphabet of the genetic code of length 4. The alphabet $N-$area is substituted by the coefficients of the corresponding polynomial p_k (special polynomial private of its constant term) cf. [13,20] writing in decimal basis then converted in binary to be used by the systems of information storage on computer. Then, it is converted into genetic code after passing through the basis 4. We have verified that all code-words are not the beginning of another code and the length of code-words of symbols less frequents have the same lengths; the results of Shannon cf. [4] and [3] on the effectiveness of the code, that is $L \geq \frac{H(S)}{\log_4(N)}$ where L is the average length of the code, $H(S)$ the entropy of the source S and $N = card(A)$; the illegality of Kraft for only encodable and instantaneous, that is $\sum_{i=1}^{N} \frac{1}{4^{l_i}} < 1$ where l_i the length of the code-word C_i. We get a prefix, optimal, effective, high-performance and without error code. We find that the coding system presents an injectivity gain. We created a encoding and decoding software which will be put in the annex for the application. A possible

improvement of the software will be considered for resolving certain problem, that is reduce as much as possible the application numbers for this coding type.

2. PRELIMINARY

2.1. **Special polynomial.**

Theorem 2.1. *[5] There is an unique polynomial $f_P(n)$ of degree $dim(P)$ with rational coefficients such that: $f_P(n) = card\left(nP \cap \mathbb{Z}^d\right)$, for all $n \geq 1$, $d \geq 2$ dimension of network $\mathbb{Z}$. In addition we have $f_P(0) = 1$, $f_P(-n) = (-1)^{dim(P)} card\left(nP^0 \cap \mathbb{Z}^d\right)$, for all $n \geq 1$ (Reciprocity law).*

Proof. See [5]. □

Remark 2.1. *The Ehrhart polynomial $f_P(n)$ of an integer convex polytope P of d dimension can also be written under the form : $f_P(n) = c_d n^d + \cdots + c_1 n + c_0$ where $c_0 = 1$.*

Definition 2.1. *[13, 20] We call family of Ehrhart polynomials the polynomial that we write: $g_{m,d,k}(n) = \prod_{j=d-k+1}^{d}(n+j) + m\prod_{j=0}^{k-1}(n-j)$, with $k = \left[\frac{d+1}{2}\right] \in \mathbb{N}^\star$ (degree), $m \geq 0$ (parameter) and $(d \geq 2)$ (dimension of P).*

Definition 2.2. *[20] We call special polynomial denoted by p_s the polynomial derived from the family of Ehrhart polynomials $g_{m,d,k}(n)$ that we can write under the form:*

$$p_s(n) = \sum_{i=0}^{k-1} c_{k,i} n^{k-i} + a_0$$

with $J_d \cong a_0 \mod (d+k)$.

We denote $p_k(n)$ the special polynomial without constant term in which the coefficients are the decomposition factors of the constant term J_d of $g_{m,d,k}(n)$ in product of decreasing factors. We write

$$p_k(n) = \sum_{i=0}^{k-1} c_{k,i} n^{k-i} = p_s(n) - a_0$$

and $J_d = \prod_{j=d-k+1}^{d}(j) = \prod_{i=0}^{k-1}(c_{k,i})$.

Example 1. *For $d = 7$ we have $k = \frac{7+1}{2} = 4$, $J_7 = (7)(6)(5)(4) = 840$ and $p_s(n) = p_k(n) + a_0$ where $p_k(n) = 7n^4 + 6n^3 + 5n^2 + 4n$ with $a_0 = 4$.*

2.2. **Genetic code [9, 16].** The support of the genetic information is the DNA (Desoxyribonucleic acid). It is the basis of the heredity: phenomenon of transmitting of the genetic information of a mother cell to the daughter cells. Every cell contains all the genetic code but the genes are expressed differently from one cell to another. As for the various operations concerning the copies of coding or the one-to-one translation, completed from the nucleic acid, we get the following steps:

- **replication**: decoding of a DNA to produce a complementary DNA in accordance with the following rule:

$$\text{Adenine}\,(A) \longrightarrow \text{Tymine}\,(T)\,; \qquad (T) \longrightarrow (A)\,;$$

$$\text{Guanine}\,(G) \longrightarrow \text{Cytosine}\,(C)\,; \qquad (C) \longrightarrow (G)\,.$$

- **transcription**:decoding of an DNA for product an RNA (Ribonucleic acid) according to rule:

$$(A) \longrightarrow \text{Uracil}\,(U)\,; \qquad (T) \longrightarrow (A)\,; \qquad (G) \longrightarrow (C)\,; \qquad (C) \longrightarrow (G)\,.$$

- **translation**:decoding of a RNA messager to produce a protein, the rules of this translation form the genetic code.

We have the following table of complementarity [16]:

$$RNA \text{ transferts } (RNA_t) \longrightarrow RNA \text{ messager} (RNA_m)$$

TABLE 1. Table of complementarity

ARN_t	A	C	G	U
ARN_m	U	G	C	A

Source: [16]
For the DNA: $T,\ A,\ C,\ G$ we have 4 basis and the same for RNA: $U,\ A,\ C,\ G$. Then they have all the same number of the 4 basis in the numeration. In addition, our encoding is part of the entropic coding or reversible compression of a source corresponding to a coding without loss of symbols.

2.3. **Entropic coding [18, 21].** The notion of entropic encoding is fundamental into the theory of codes, namely the inequality of Kraft [8], the theorem of Shannon [22].

Definition 2.3. *Let S be a source defined by its alphabet symbols $\{s_1, \ldots, s_N\}$ and its emission characteristics governed by a probability law $P := \{p(s_1), \ldots, p(s_N)\}$. A source is simple (or without memory) if the symbols emited by the source S are independent and of the same law.*

A sequence of N symbols emited at instants $1, 2, \ldots, N$ follows a probability law $P(s_1, s_2, \ldots, s_N) = p(s_1)\, p(s_2) \ldots p(s_N)$.

The entropy of zero order $H(S)$ of a simple source S, of the probability law P is defined by $H(S) = -K \sum_{i=1}^{N} p(s_i) \log_2 [p(s_i)]$ where K is positive constante cf. [10, 18]. Taking into account K as unity of measure, we have:

$$H(S) = -\sum_{i=1}^{N} p(s_i) \log_2 [p(s_i)] .$$

Property 1. *The entropy $H(S)$ is maximal if all symbols $\{s_1, \ldots, s_N\}$ of the source S are equiprobables. The entropy is then equal to the information associated to each message taken individually. Then for all $i \in \{1, \ldots, N\}$, $p(s_i) = \frac{1}{N} \Longleftrightarrow H(S) = \log_2(N)$.*

Proof. Immediate. □

Inequality of Kraft 1. *It makes up a basic result in theory of codes. It provides a necessary-sufficient condition of existence of instantaneous decipherable codes expressed in function of the length of code words. This inequality is written: $\sum_{i=1}^{N} \frac{1}{2^{l_i}} \leq 1$ where l_i (with $i \in [1, \ldots, N]$) the length of candidate-words for code a source $N-$area in a binary alphabet [8].*

Theorem of Shannon 1. *Let us consider a without memory stationary source, the average length $\overline{n}$ of coded words m_i is limited by the entropy value of the source S,*

$$\overline{n} = \sum_{i=1}^{N} l_i p(s_i) \geq \sum_{i=1}^{N} p(s_i) \log_2 [p(s_i)] .$$

It is also possible to find a decipherable code such that $H(S) \leq \overline{n} < H(S) + 1$. The average length of a symbol is given by the formula $\alpha = \sum_{i=1}^{N} p_i l_i$ (bits/symbol).

3. Application of the special polynomial to the encoding

3.1. Construction of the code: Principle basis.

Let's consider a source S defined by its alphabet $A = \{s_1,\ s_2, \ldots,\ s_N\}$ of N symbols and, its emission characteristics are given by a probability law $P := \{p(s_1), p(s_2), \ldots, p(s_N)\}$.

Every symbol of the source will be emitted at the instants $t_1, \ldots, t_n$ corresponding to its rank in the message. The total number of coefficients of the special polynomial p_k must be equal to the total number of symbols in the alphabet $N-$area and its degree is $k = N = \left[\frac{d+1}{2}\right]$, k integer, $k \geq 1$. We identify every coefficient of p_k for each symbole of the alphabet A of the most great coefficient to the first rank in the alphabet of the source, the next is second rank, and so on, that is the coefficients by decreasing greatness order and the symbols by increasing classification.

We use 137 characters composed of letters and symbols which appear in the keyboard of a "standard" computer and the 4 length figure of the genetic code. For each character, we associate a number between 0 and 137.
Since $4^3 = 64 < 137 \leq 254 = 4^4$, then we can do the coding at least on 4 bits. Since we adopt a code, let's say entropicogenetic of variable length strictly superior than the length of the genetic code which equals to 4 in conformity with the conception of any code of variable length using the representation in the form of tree.

3.2. **Method: Algorithm of dynamic progression.**

The algorithm consists of the following steps:

Step 1:

(i) Write and number the alphabet constituating the message by order in the source, including the punctuations and the white space, which are denoted by A in this set. The ponctuations and the space don't appear in A but have of number according to their classification.

(ii) Write the corresponding polynomial p_k the degree k of which is equal to N (number of alphabets) and identify to A,

$$p_k(n) = \sum_{i=0}^{k-1} \binom{n^{k-i}}{k, i} = \binom{n^k}{k, 0} + \binom{n^{k-1}}{k, 1} + \cdots + \binom{n}{k, k-1},$$

$$\left(\binom{}{k,0},\ \binom{}{k,1}, \ldots,\ \binom{}{k,k-1}\right) = (s_1,\ s_2, \ldots,\ s_N).$$

(iii) Define the first application: $\varphi_1 : A \longrightarrow C_1$ such that $\varphi_1(s_i) = C_{k,j}$, with $A := \{s_1, \ldots, s_N\}$,

$C_1 = \{c_{k,0}, \ldots, c_{k,k-1}\}$, $i \in [1, N]$, $j \in [0, k-1]$. In other words,

$$\begin{cases} \varphi_1(s_1) = C_{k,0} \\ \varphi_1(s_2) = C_{k,1} \\ \vdots \\ \varphi_1(s_N) = C_{k,k-1} \end{cases}.$$

Step 2:
Let us denote by φ_2 the 2^{rd} application, o the order of the alphabet (from 101 and r the rank or classification in the source), $C_2 = \{s_{ior} : i \in [1, N], o \geq 101,\ r \geq 1\}$. The application φ_2 is then defined by $\varphi_2 : C_1 \longrightarrow C_2$ such that $\varphi_2(C_{k,j}) = s_{ior}$, that is

$$\begin{cases} \varphi_2(C_{k,0}) = s_{1o1} \\ \varphi_2(C_{k,1}) = s_{1o2} \\ \vdots \\ \varphi_2(C_{k,k-1}) = s_{NoN} \end{cases}.$$

Step 3:
Let us define the 3^{rd} application. Let be $\varphi_3 : C_2 \longrightarrow C_3$ such that s_{ior} converted in basis 4. We denote s_{ior4} the number s_{ior} in basis 4, then $C_2 := \{s_{ior4} : i \in [1, n], o \geq 101, r \geq 1\}$. In other words, $\varphi_3 : C_2 \longrightarrow C_3$ such that $\varphi_3(s_{ior}) = s_{ior4}$:

$$\begin{cases} \varphi_3(s_{1o1}) = s_{1o14} \\ \varphi_3(s_{1o2}) = s_{1o24} \\ \vdots \\ \varphi_3(s_{NoN}) = s_{NoN4} \end{cases}.$$

Step 4:
The 4^{th} application φ_4 is defined as follow: $\varphi_4 : C_3 \longrightarrow C_4$ where $C_4 = \{(s_{ior4})$ converted in $DNA\}$ according to the below table.

TABLE 2. correlation table

Number	0	1	2	3
ADN	A	C	G	T

Source: Author
We denote $(s_{ior4a}) = (s_{ior4})$ converted in DNA according to the above table. $\varphi_4 : C_3 \longrightarrow C_4$ such that $\varphi_4(s_{ior4}) = s_{ior4a}$.

Step 5:
The last application is such that $\varphi_5 : C_4 \longrightarrow C_5$. By transcription and translation following the rules of the genetic code (decoding of an DNA to produce an RNA and endecoding of an RNA messager to produce a protein). We denote $(s_{ior4a})_{t-m} \in C_5$ converted in $RNA_t \rightarrow RNA_m$ according to the complementarity table.

TABLE 3. Translation and transcription

DNA	A	C	G	T
RNA_t	U	G	C	A
RNA_m	A	C	G	U

Source: [16]

Remark 3.1. *The numbers or orders of characters or symbols start from the figure* 101 *to avoid the restriction of the message length.*

3.3. **Reminder.**

Definition 3.1. *[1] Let* $\Phi : \{0,1\}^k \longrightarrow \{0,1\}^n$ *be an application. A code is an injective application, that is all element of the arrival set has at most an antecedent in the departure set.* Φ *is a binary code with length* n *and of dimension* k *.*

Definition 3.2. *[14] A code is said prefixe if none of the code word start by another code word.*

Proposition 3.1. *The application* $\varphi_1 : A \longrightarrow C_1$ *is a code such that* $\varphi_1(s_i) = c_{k,j}$ *where* $A = \{s_1, \ldots, s_N\}$, $C_1 = \{c_{k,0}, \ldots, c_{k,j}, \ldots, c_{k,k-1}$ *with* $i \in [1, \ldots, N]$, $\quad j \in [0, \ldots, k-1]$ *with* $j = i - 1$.

Proof. Let N be the number of symbols of the alphabet A, C_1 the set of coefficients of p_k, $card(C_1) = \deg(p_k(n)) = N$. For all symbol of the source is associated one and only one coefficient of p_k, its image. So φ_1 is a bijective application from the departure set A to the arrival set C_1, φ_1 is then injective hence φ_1 is a code. □

Similarly for the applications φ_2, φ_3, φ_4 and φ_5. Since the composed of bijective applications is a bijection then the composed of codes form a code.

Proposition 3.2. *The entropicogenetic encoding allows us to have a prefixe code.*

Proof. This is immediate using the previous Definition 3.2. □

In a code with variable length (VLC) cf. [6], we can do a compression without loss [10] if the code is prefixe and the symbols aren't equiprobables.

In a compression system without loss cf. [14], the decodor is capable of rebuilding exactly the source datas (case of prefixe code).

An another method of compression: the method of compression RLE see [17], [7] used by many formats (BMP, PCX, TIF) in which the basis principle consists to code a first element giving the number of repetition of a value then completing it by the value to be repeated. In fact the compression RLE allows us to compress when this is necessary and to leave the chain as it is when the compression induces a waste (gain of negative decompression). If three elements or plus are repeated consecutively then the method RLE is used. For example: Chain AAAAAHHHHH-HHHHHHHHH, compressed, composed by two different lettres gives 5A14H. The compression gain is then $\frac{19-5}{19} = 73,7\%$.

Proposition 3.3. *In the entropicogenetic code we can do a compression without loss,for all* $p_k,\ k \geq 1$ *integer.*

Proof. Since the entropicogenetic code is a prefixe code and if in addition the symbols aren't equiprobable then we can do a compression without loss according to [6]. □

Definition 3.3. *[3,4] A code is said to be a unique endecoding if its associated coding is injective.*

Lemma 3.1. *The entropicogenetic code is of unique decoding.*

Proof. This is immediate according to the extension of the previous Proposition 3.1. □

Theorem 3.1. *[15] The lengths of code-words of a code* $N-$*area only decodable must satisfy the inequality of Kraft.*

Theorem 3.2. *The entropicogenetic encoding allows to have a more high-performance code than an only decodable code. Besides, all entropicogenetic code is effective.*

Proof. According to the Proposition 3.2 the entropicogenetic code is prefixe code, that is an instantaneous code so all is also high-performance in length. But the

Lemma 3.1 ensures us the difference with the only encodable code [24]. Let S be a source of a quaternary code of length L, $H(S)$ its entropy. The codes are of variable length. By the theorem of Shannon's encoding we can determine what's the number of bits of the information used in order for the code to be effective, $L \geq \frac{H(S)}{\log_4 N}$ where N points out the number of the alphabet symbols of the source, that is, the source length of this encoding must be superior or equal to the division between the value of the source entropy and the $\log_4 N$. □

Theorem 3.3. *[21] In order to have a coding without error, a source S must be coded in average with at least $H(S)$ bits, that is, $L \geq H(S)$.*

Theorem 3.4. *The entropicogenetic encoding is a coding without error.*

Proof. We just need to use the theorem 3.2 and the theorem 3.3. □

Proposition 3.4. *In the entropicogenetic code, the probabilities of apparition of symbols are not always equiprobable.*

Proof. We suppose that in a message of the source S, there are symbols having different frequencies. In other words, there is one or several repetitions of symbols. Then the most frequent symbols have higher probability than the ones less frequent.

In fact, the symbols don't always have the same probabilities of apparition. In the opposite case then, there is an equiprobability, that is, all symbols have the same frequency of apparition [19]. □

Corollary 3.1. *Let S be a source such that $S = \{s_1, \ldots, s_N\}$ with $p(s_i)$ the probability of apparition of the symbol s_i, and l_i the length of the code c_i where $i = 1, \ldots, N$. In the entropicogenetic code if there exists $j \neq i$ and $p(s_i) > p(s_j)$ then $l_i > l_j$. In addition, if we have $p(s_i) = p(s_j)$ then $l_i = l_j$.*

Proof. This is from the previous proposition 3.4. □

Example 2. *To illustrate the coding algorithm, we will code the following sentence: "virer à droite".*

***Step** 1: By order of apparition in the message, the source S consists of $N = 8$ symbols $= k = deg^{\circ} p_k(n)$ in which the corresponding alphabet A is that $A = \{v, i, r, e, , d, o, t\}$. The total number of characters emanating from the source S is equal to 12. The following table gives the letters of the alphabet A, the frequence for each symbol and the probability of its apparition.*

TABLE 4. Probability of occurrence for each symbol

Symbols	*Number of times*	*Probability*
v	*1*	$1/12$
i	*2*	$2/12$
r	*3*	$3/12$
e	*2*	$2/12$
à	*1*	$1/12$
d	*1*	$1/12$
o	*1*	$1/12$
t	*1*	$1/12$

***Source**: Author*

Let us write the polynomial $p_k(n)$ associated to the previous symbols of the message: $k = 8 = \frac{d+1}{12}$ this gives $d = 15$ then $p_k(n) = 15n^8 + 14n^7 + 13n^6 + 12n^5 + 11n^4 + 10n^3 + 9n^2 + 8n$.

The first application is given $\varphi_1 : A \longrightarrow C_1$ where $C_1 = \{15, 14, 13, 12, 11, 10, 9, 8\}$

$$\begin{cases} \varphi_1(v) = 15 \\ \varphi_1(i) = 14 \\ \varphi_1(r) = 13 \\ \varphi_1(e) = 12 \\ \varphi_1(\grave{a}) = 11 \\ \varphi_1(d) = 10 \\ \varphi_1(o) = 9 \\ \varphi_1(t) = 8 \end{cases}$$

We will get the successive applications of steps 2, 3, 4 and 5.

***Step** 2: We have the 2^{nd} application $\varphi_2 : C_1 \longrightarrow C_2$*

$$\begin{cases} \varphi_2(15) = 17701 \\ \varphi_2(14) = 1330212 \\ \varphi_2(13) = 163030510 \\ \varphi_2(12) = 1190414 \\ \varphi_2(11) = 10307 \\ \varphi_2(10) = 11709 \\ \varphi_2(9) = 15111 \\ \varphi_2(8) = 16713 \end{cases}$$

Step 3*: We obtain the* 3^{rd} *application* $\varphi_3 : C_2 \longrightarrow C_3$ *(conversion in basis 4)*

$$\begin{cases} \varphi_3(17601) = 10110211 \\ \varphi_3(1320212) = 11010300210 \\ \varphi_3(162030510) = 21231322113232 \\ \varphi_3(1190414) = 10202220032 \\ \varphi_3(10307) = 2201003 \\ \varphi_3(11709) = 2312331 \\ \varphi_3(15011) = 3230013 \\ \varphi_3(16613) = 10011021 \end{cases}$$

Step 4*:We have the* 4^{th} *application* $\varphi_4 : C_3 \longrightarrow C_4$ *(conversion in* DNA *according to the table 2)*

$$\begin{cases} \varphi_4(10103001) = CACCAGCC \\ \varphi_4(11002110110) = CCACATAAGCA \\ \varphi_4(21222012032232) = GCGTCTGGCCTGTG \\ \varphi_4(10202220032) = CAGAGGGAATG \\ \varphi_4(2201003) = GGACAAT \\ \varphi_4(2312331) = GTCGTTC \\ \varphi_4(3222203) = TGTAACT \\ \varphi_4(10003211) = CAACCAGC \end{cases}$$

Step 5*:We get the* 5^{th} *application* $\varphi_5 : C_4 \longrightarrow C_5$ *(conversion in* RNA_t *then* RNA_m*)*

$$\begin{cases} \varphi_5(CACATAAC) = CACCAGCC \\ \varphi_5(CCAAGCCACCA) = CCACAUAAGCA \\ \varphi_5(GCGGGACGATGGTG) = GCGUCUGGCCUGUG \\ \varphi_5(CAGAGGGAATG) = CAGAGGGAAUG \\ \varphi_5(GGACAAT) = GGACAAU \\ \varphi_5(GTCGTTC) = GUCGUUC \\ \varphi_5(TGGGGAT) = UGUAACU \\ \varphi_5(CAAATGCC) = CAACCAGC \end{cases}$$

Finaly, we get the entropicogenetic code of symbols emanating from the alphabet A of the source S. Let us calculate the entropy of the source S. First, the average length of a symbol $L = \sum_{i=1}^{8} p_i l_i = 10,166$ bits by symbol. Next the entropy of the source $H(S) = -\sum_{i=1}^{8} p_i \log_4 p_i = 1,42$. Since $10,16 > 1,42$ then the inequality in the basic therem of Shannon is verified. There are more compression:

8 symbols of the alphabet A with less 2 bits, that is for coding $4^2 = 16$ different messages, we use 2 bits (in basis 4). In addition, the Kraft inequality is also verified. Moreover the code is optimal because the two less frequent symbols have the same length, that is, $p(i) = p(e) = \frac{2}{12}$ then $l_i = l_e$. Apart from that we notice that none of the word of this code start by another code-word so it is a prefixe code, that is instantaneous code, consequently this is an high-performance code. Then this is immediate to verify that this code is also effective. Finally, by applying the first theorem of Shannon, we find that $L \geq H(S)$ therefore the coding is without error, hence the authenticity of this coding.

3.4. **Application.** For the application of the entropicogenetic encoding and decoding we have created a software. We would like to invite our readers to consult the librairy mentioned in the annex.

4. Discussion and conclusion

The coding that we have just proposed is a combination of entropy and genetic encoding, that we call "entropicogenetic coding". Entropic encoding because this is prefix code based on the theory of Shannon, the average information of symbols of the source, that is the entropy of the source and, genetic encoding since we use the genetic alphabet then the genetic code of length 4.

We use an algebraic approach: We use a special polynomial $p_k(n) = \sum_{i=0}^{k-1} c_{k,i} n^{k-i}$ with degree equal to the number of elements of the alphabet derived from the source, that is $\deg[p_k(n)] = card(A)$ where A points out the alphabet of the source S; the algebraic property of the function $\varphi_i,\ i \in [1,5]$ (bijectively and composition of bijections). The algorithm of coding is done in five (5) steps: from alphabet $N-$area of $A = \{s_1, \ldots, s_N\}$ to a quaternary alphabet $C_5 = \{A, C, G, U\}$ alphabet of the genetic code. We have had a code verifying the fundamental theorem of Shannon and the inequality of Kraft.

This is a prefix, optimal, effective and high-performance code with a compression without loss and authentic. This essentially depends on the source of the message to be transmitted. An encoding and decoding software have been created for the application (see library in annex). For this, we would like to bring in the future some improvements: First, we will reduce as much as possible the number of process steps of encoding (number of applications), that is we will simplify the encoding algorithm. Then we will make sure that there is not limit in alphabet of

any nationalities (for example: Chinese, Greek,...), then we will create a software adapted to the change of characters and outside of the computer keyboard which will be used in numerical world in accordance with the current development of technology.

APPENDIX A. IMPLEMENTATION OF THE CODING METHOD [11, 12]

In order to be able to exploit and use our method in different applications, we have created a library coded in JAVA in $C\#$ because of that these program languages are very strong, very used and widespread in the technology fields. These libraries use the ”BigInteger” class and the librairy ”gwt-math” for JAVA [11] and ”numerics” for $C\#$ [12] that allows us to represent the integers without any size limitations. We can then store and calculate the very big numbers during the treatment of message. The calculations then are not limited to the character numbers of the message, which makes it easier for the librairy to encode even those very long message. All of that is explained in the section below.

APPENDIX B. DESCRIPTION OF FUNCTIONALITIES OF THE LIBRARY

Our librairy contains different functions that we can use during its use. We will see the description of these functionalities and the constructors in the next pages.

TABLE 5. Summary of the constructor

Constructors **Constructor and description**
Entropic Codage() Build a new codor-decodor.
Entropic Codage(String mes) Create a codor whose message to be encrypted is the message "mes".

Table 6: Summary of the method

Modificator and Type	**Method and description**
public String	**encodageSymetrique(String sms)** Send back the encrypted message associated to the message "sms"

public String	**decodageSymetrique (String sms)** send back the initial message associated to the encrypted "sms".
public String	**encodageAsymetrique(String sms)** Send back the encrypted message associated to the message "sms" and secured by a private key.
public String	**getClePriveAsymetrique()** Send back the private key for an asymetric encryption.
public String	**decodageASymetrique(String sms, String cle)** Send back the initial message associated to the encrypted message "sms" by means of the private key "cle".
public void	**rootCryptage()** Send back the encrypting steps.
public void	**rootDecryptage()** Send back the decrypting steps.

APPENDIX C. CHARACTER LIST WITH THEIR NUMBER

The table below shows the list of characters appearing in the keyboard of nowdays's computer with their respective corresponding numbers used during the encryption.

TABLE 7. list of characters

101 a	102 A	103 à	104 À	105 â	106 Â	107 ä	108 Ä	109 ã	110 Ã
111 b	112 B	113 c	114 C	115 ç	116 Ç	117 d	118 D	119 e	120 E
121 é	122 É	123 è	124 È	125 ê	126 Ê	127 f	128 F	129 g	130 G
131 h	132 H	133 i	134 I	135 î	136 Î	137 ï	138 Ï	139 j	140 J
141 k	142 K	143 l	144 L	145 m	146 M	147 n	148 N	149 ñ	150 Ñ
151 o	152 O	153 ò	154 Ò	155 ô	156 Ô	157 õ	158 Õ	159 p	160 P
161 q	162 Q	163 r	164 R	165 s	166 S	167 t	168 T	169 u	170 U
171 ù	172 Ù	173 û	174 Û	175 ü	176 Ü	177 v	178 V	179 w	180 W
181 x	182 X	183 y	184 Y	185 ÿ	186 Ÿ	187 z	188 Z	189 &	190 "
191 #	192 '	193 {	194 (	195 [	196 -	197 |	198 ∘	199 +	200 _

201 \	202 @	203)	204]	205 =	206 }	207 £	208 $	209 ¤	210 μ
211 *	212 %	213 ?	214 ,	215 .	216 ;	217 :	218 /	219 §	220 !
221 ^{2}	222 1	223 2	224 3	225 4	226 5	227 6	228 7	229 8	230 9
231 0	232 ^	233 ¨	234 ‘	235 ˜	236 <	237 >			

REFERENCES

[1] P. ABBRUGIATI: *Introduction aux codes correcteurs d'erreurs, Chap 2*, Université de Nice, (12 Novembre 2008).

[2] J. ANDRÉ: *Caractères, Codage et normalisation de Chappe a Unicode*, Document numérique, Editions Lavoisier + Hermès, **6**(3,4) (2002), 13–49.

[3] M.-P. BÉAL: *Codage Symbolique*, Masson, 1993.

[4] M.-P. BÉAL, N. SENDRIER: *Théorie de l'information et codage*, Université de Marne-la-Vallée, France, (21 nov 2012), 67 pages.

[5] M. BRION: *Points entiers dans les polytopes convexes*, Exposé dans le Séminaire Bourbaki (1993/1994). Astérisque **227**(780) (1995), 145–169

[6] M. CAGNAZZO: *Principes du codage sans perte*, Département Traitement du Signal et des Images TELECOM ParisTech, 2013.

[7] V. CHAPPELIER: *Codage progressif d'images par ondelettes orientées*, Thèse de Doctorat, Université de Rennes 1, 2015, 223 pages.

[8] T. M. COVER, J. A. THOMAS: *Elements of information theory*, Wiley, 1991.

[9] *Cours IFT 3290*, Démo 2: (le 27 Janvier 2003).

[10] A. GARIVIER: *Codage et entropie métrique : l'exemple des classes enveloppes*, Réunion de rentrée de lequipe MAFIA Université Paul Sabatier - Toulouse.

[11] J. GOSLING, D. C. HOLMES, K. ARNOLD: *The Java programming language*, 2005.

[12] A. HEJLSBERG, S. WILTAMUTH, P. GOLDE: *C# language specification*, Addison-Wesley Longman Publishing Co., Inc., 2003.

[13] A. HIGASHITANI: *Counterexamples of the conjecture on roots of Ehrhart polynomials*, Discrete comput Geom, **47** (2012), 618–620.

[14] P. JARDIN: *Codage de source*, ESIEE, (03 octobre 2008).

[15] M. LELARGE: *Théorie de l'information et codage- Cours 2*, Scribe: Paul Simon, (22 Février 2011).

[16] N. EL-MABROUK: *Introduction: le code génétique*, DIRO Université de Montréal; Inspiré de " An Introduction to Bioinformatics Algorithms", 2015.

[17] B. MARTIN: *Codage, cryptologie et applicatios*, Russe Polytechnique, et Universitaires Romandes, 2004, 354 pages.

[18] N.J. MVOGO: *Cours compression d'images- Master II: IASIG: Principes Généraux de Codage entropique d'une source*,

[19] D. PETRITIS: *Probabilités pour la théorie de l'information- Notes de cours Rennes*, UFR Mathématiques, Université de Rennes 1, 2015, 105 pages.

[20] J. J. RAKOTO, H. S .G. RAVELONIRINA: *On family of Ehrhart polynomials and Counterexamples of the conjecture of Beck and al.*, Annals of Pure and Applied Mathematics, **10**N. 2 (2015), 285–299.

[21] R. RHOUMA: *Théorie de l'information- Chap I: Codage source*, Ecole Supérieure d'Economie Electronique, (Avril 2015). https:// sites.google.com/site/rhoouma.

[22] O. RIOUL: *Théorie de l'information et du codage*, Hermes Sciences Lavoisier, 2007.

[23] P. E. I. SALL: *Architecture des machines numériques: Décimal codé binaire*, Université du Sahel, 2019/10.

[24] B. TORRÉSANI: *Codage et Compression des Signaux, Partie II-Cours de DESS 2003-04 Marseille*, Université de Provence Marseille, 2003, 103 pages.

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
UNIVERSITY OF ANTANANARIVO, ANTANANARIVO 101- MADAGASCAR
Email address: rhsammy@yahoo.fr

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
UNIVERSITY OF ANTANANARIVO, ANTANANARIVO 101- MADAGASCAR
Email address: rjeanjolly@gmail.com

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
UNIVERSITY OF ANTANANARIVO, ANTANANARIVO 101- MADAGASCAR
Email address: razakasoaharymahefa@gmail.com