Metrics for Symbol Clustering from a Pseudoergodic Information Source

Angel F. Kuri Morales
Instituto Autónomo de México
Río Hondo No. 1 México D.F.
akuri@rhon.itam.mx

Oscar Herrera Alcántara
Centro de Investigación en Computación
Av. Juan de Dios Batiz s/n México D. F.
heoscar@yahoo.com

Abstract

We discuss a set of metrics, which aims to facilitate the formation of symbol groups from a pseudoergodic information source. An optimal codification can then be applied on the symbols(such as Huffman Codes [1]) for zero memory sources where it tends to the theorical limit of compression limited by the entropy. These metrics can be used as a fitness measure of the individuals in the Vasconcelos genetic algorithm as an alternative to exhaustive search.

Keywords. Metrics, information source, codification, entropy, genetic algorithm.

1. Introduction

In the original work of Shannon [2] the concept of amount of information assigned to a symbol is defined. Such symbol is generated by a source, which is assumed to be unknown but from which we can extract its probability. So, the associated information to that symbol can be defined as:

$$I(s_i) = -\log(pi) = \log(\frac{1}{p_i})$$

where s_i denotes the i-th symbol and p_i denotes its probability. This definition satisfies two intuitive characteristics about information. First, it associates the most information to that symbol which is more unexpected, i.e., symbols which do not often occur provide more information than symbols which often do. Second, it encloses the idea that information must be additive, i.e. the information of two symbols must be equal to the sum of both of them.

$$I(s_1) + I(s_2) = I(s_1, s_2)$$
$$= \log\left(\frac{1}{p1,2}\right) = \log\left(\frac{1}{p1p2}\right)$$
$$= \log\frac{1}{p1} + \log\frac{1}{p2}$$

This exhibits, clearly, the probabilities as a product and the amount of information as a sum. The basic concept associated to that definition is the average information of the source named entropy denoted by:

$$H(x) = \sum p_i \log\left(\frac{1}{p_i}\right)$$

The above discussion tacitly assumes the statistical independence of the symbols. In fact, all the Classic Information Theory [3] is based on the fact that source information is ergodic. We say a process is ergodic if we can pass from a state to other and whenever $t \to \infty$, the system reaches stabilization in an independent distribution from the initial state. In the practice, however, sources are not ergodic. In [2] Shannon discuses a set of English approximations series which he called first, second and third order to independent "symbols" and first and second order to English "words". These two (arbitrary) choices are naturally desired. A symbol can be easily identified because it exists as a physical entity, while a "word" is delimited by spaces between symbols. If there were statistical independence, as we point out, the election of n will be arbitrary, but Shannon affirmed [4]: "Rather than continue with tetragram, , *n*-gram structure it is easier and better to jump at this point to word units". This affirmation reflects the nonergodicity of the information sources which we must often treat.

2. Metasymbols

We propose to identify groups of symbols (metasymbols) that, when optimally encoded, can give an approach to data compression. A metasymbol contains information about a symbol set not necessarily adjacent within a message. For example, given a message msg = "abcdefghijklm", a decomposition in metasymbols would be: M={abm, ef, d, ghijk, lc}. The grouping of nonadjacent symbols can be conceived as if

the symbols moved with respect to their original position (index) within the message. From this point of view, the idea to move the symbols of its original position to codify them in an optimal way is not new, the Burrows-Wheeler Transform [5] is an example of this. Therefore, some permutation of the symbols in a string promotes compression of the complete message composed by strings later using a compression technique like Move To Front [6].

The necessary number of metasymbols to codify a message is denoted by |M|, where $M = \{\alpha, \beta, \chi, \delta, \varepsilon, ...\}$.

In what follows we use Greek letters to represent metasymbols. The position of the symbols in the original message appears like a subscript.

By definition, a metasymbol is considered different from another one if:

a) The constituent symbols differ from those of any other metasymbol.

Example: $\alpha = a_1 b_2 c_3$

$\beta = a_4 e_5 f_6$

b) They differ in the relative positions (relative to the first symbol) of the symbols that compose them.

Example:

Let the message, msg ="$a_0 a_1 b_2 a_3 c_4 a_5 a_6 a_7$"

χ =	a_0	a_1	a_3
Absolute positions in the message msg.	0	1	3
Relative positions to the first symbol of χ.	0	1	3

δ =	a_5	a_6	a_7
Absolute positions in the message msg.	5	6	7
Relative positions to the first symbol of δ.	0	1	2

c) They differ in the number of symbols that constitute the metasymbol, called in what follows, length of the metasymbol.

Example:

$\alpha = a_0 b_1 c_2$

$\beta = a_5 b_4 c_5 d_6$

$\text{length}(\alpha e = l_\alpha = 3$

$\text{length}(\beta e = l_\beta = 4$

Considering the metasymbols group, for nonadjacent symbols we introduce a special symbol *, meaning "lack of length" which allows us to fill up the spaces between the symbols of the message. For example, let the message be msg="abcdefghijklm". We can have the metasymbols shown in the table 1.

Table 1. Some metasymbols and their lengths for the message "abcdefghijklm"

Metasymbol	Length
$\alpha = a_0 b_1 c_2$	3
$\beta = d_3 e_4 f_5 ** i_8 j_9 * l_1$	6
$\chi = g_6 h_7 ** k_{10} * m_{12}$	4

Symbol groups are chosen to build metasymbols which are independents from each other, hoping that, a codification like Huffman Coding [7] on the metasymbols provides better compression than when symbols were coded assuming each symbol it precedes was independent from it, i.e., we eliminated first order ergodic presumption of the source.

Some questions arise when we deal with groups of symbol, such as:

¿How many groups to build?

¿How many symbols belong each group?

¿What symbols belong to a group?

¿How redundant must be a group?

Motivated by the previous questions we have analyzed several metrics which are discussed next.

3. Metrics for symbol clustering

Some ideas about symbol clustering in a message with finite length L are:

- If groups have length close to L, then there are not many repetitions of groups. Besides, in the worst case, the message is decomposed in just one group equal to the original message and that case is to be avoided.
- If groups have minimal length close to 1, "metasymbols" are reduced to the original symbols and that case is also to be avoided.
- To promote compression, there must exist groups that repeat and they may be found looking for patterns of coincidence between symbols and its positions.
- A low number of metasymbols is desired in order to diminish the dictionary size in a codification like Huffman Coding.

The last ideas may be resumed in three essential points:

1. Diminish the number of metasymbols |M|.
2. Maximize the length of each metasymbol, in order to favor (1).
3. Maximize the frequency of appearance of each metasymbol to promote compression.

To this effect we have proposed the metrics shown in the table 2.

Table 2. Metrics for symbol clustering

$F_1 = \sum_{i=1}^{\lvert M\rvert} (p_{mi} \log \frac{1}{p_{mi}})$	(1)
$F_2 = \sum_{i=1}^{\lvert M\rvert} \frac{(p_{mi} \log \frac{1}{p_{mi}})}{l_{m_i}}$	(2)
$F_3 = \lvert M \rvert \sum_{i=1}^{\lvert M\rvert} (p_{mi} \log \frac{1}{p_{mi}})$	(3)
$F_4 = \lvert M \rvert \sum_{i=1}^{\lvert M\rvert} \frac{(p_{mi} \log \frac{1}{p_{mi}})}{l_{m_i}}$	(4)
$F_5 = \frac{\left[\sum_{i=1}^{\lvert M\rvert} (p_{mi} \log \frac{1}{p_{mi}})\right] - D}{W}$	(5)

where :

F_i The discrimination measurement represent different possibilities of classification

p_{m_i} Metasymbol probability.
The number of times the metasymbol m_i appears (i.e. f_{m_i}) when we code the message, divided by N, the total number of groups in which the message was divided. $p_{m_i} = \frac{f_{m_i}}{N}$.

l_{m_i} The length of the metasymbol.
Is the number of symbols which belongs to the metasymbol m_i **.**
The special symbol * doesn't increase l_{m_i}.

$\lvert M \rvert$ The number of metasymbols founded when we divide the message in N groups.
|M| y N are, in general, different. In fact, $\lvert M \rvert \leq N$ and the equality is given when there are no repeated groups.

$\log p_{mi}$ The amount of information associated to the metasymbol m_i.

D $D = 2 \lvert M \rvert + \log \lvert M \rvert$
It is an offset for the entropy of the message built with metasymbols. It allows us to penalize the growth of |M|.

W $W = \prod_{i=1}^{\lvert M\rvert} (\frac{l_{m_i}}{K})^{\frac{K}{f_{l_{m_i}}}} - 1$
A scaling factor that allows forming repeated groups with length not slant to extreme values (1 y L) and controlled by the factor K.

$f_{l_{m_i}}$ The appearance frequency of the lengths of the groups.

3.1 Comparing metrics

In order to arrive to an acceptable metric we tried several alternatives. In what follows we briefly describe each of the metrics we explored included the one we were successful with.

3.1.1 First metric

Metric (1) is the entropy of the message written with metasymbols and presents some deficiencies:

- Maximizing the entropy, yields group independence. The metric favors the emergence of large number of groups which are different from one another. Thus, metasymbols are equi-probable and entropy is maximum[8].
  For example: Let the message
  msg="aaabbbcccaaabbbccc"
  $M1 = a_0$
  $M_2 = b_3$
  $M_3 = c_6$
  $M_4 = a_1a_2$
  $M_5 = b_4b_5$
  $M_6 = c_7c_8$
  $M_7 = a_9a_{10}a_{11}$
  $M_8 = b_{12}b_{13}b_{14}$
  $M_9 = c_{15}c_{16}c_{17}$
- Minimizing the number of metasymbols |M|, the metric just favors one metasymbol with length equals to L, the length of the original message, i.e., $p_{m1}=1$, $l_{m1}=L$ and F=0.

3.1.2. Second metric

Metric (2) considers the length of each metasymbol as a weighting factor, in order to find the greater metasymbols. The result was poor symbol clustering because it minimize the number of metasymbols to one.

3.1.3. Third metric

Metric (3) tries to minimize the number of metasymbols by multiplying |M| with the entropy of the message

decomposed in metasymbols. The result was not satisfactory because |M| takes the number of metasymbols to 1.

3.1.4. Fourth metric

Metric (4) tried to diminish the number of metasymbols and maximize the length of the metasymbols. The result was unsuccessful because the number of metasymbols is reduced to 2 and the length is taken to L/2 but there is no repetition of the groups. If repetition of the metasymbol were found, then |M| is reduced to 1 and we have a compression ratio 2:1.

3.1.5. Fifth metric

Metric (5) considers that when we increase the length of the metasymbols, the possibility to find repetitive groups decreases and vice versa. There are two extreme cases:

a) When there is only one metasymbols, its length decreases to the minimal value (equals 1) and its frequency reaches its maximum value (equals L).
b) When the length of the only one metasymbol is maximum (equals L) the frequency of the metasymbol is one.

Neither the two last cases is desired. However, we think about the possibility that between a) and b) there is at least one point with lengths vs. frequencies such that there is a minimal number of metasymbols with the maximum length possible and in which case the codification is optimal, i.e., with maximum compression. Considering the previous ideas, we have proposed the fifth metric trying to find at least one of these points.

4. Exhaustive search

As we pointed out, the decomposition of a message in metasymbols is not unique, there are many possibilities to choose the number of metasymbols |M| and the symbols for each metasymbol.

We will see the case of L=5. We emphasize that each symbol has a position (index) in the message, for example, if msg ="a b c d e", then the symbols are s_0 =a, s_1 =b, s_2 =c, s_3 =d y s_4 =e, and its respective indices are 0,1,2,3 y 4. The relative indices in a metasymbol are calculated by subtracting the absolute position of the first symbol to the absolute position for each symbol, so the first relative index is always equals to zero.

The number of metasymbols |M| can be 1,2,3,4 and 5. Given a value for |M| there are several possibilities for the lengths of each metasymbol as we can see in the table 3.

Table 3. Possibilities for the lengths of the metasymbols when we discompose a message with length L=5

Number of de Metasymbols	Lengths	Lengths
|M|=1	5	
|M|=2	1+4	2+3
|M|=3	1+1+3	1+2+2
|M|=4	1+1+1+2	
M=5	1+1+1+1+1	

The possibilities (4+1), (3+2), (3+1+1), (2+1+2), ...,etc., are not considered because given a |M| value and a set of lengths, metasymbols explore all the different permutations between symbols. The problem can be expressed as: ¿How many possibilities are there when we divide a message with length L=5, taken all as a whole, or, one of five and four of four, or, two of five and three of three, or, one of five, one of four and three of three, etc.?

We have

$$\binom{5}{5}+\binom{5}{1}\binom{4}{4}+\binom{5}{2}\binom{3}{3}+\binom{5}{1}\binom{4}{1}\binom{3}{3}+\binom{5}{1}\binom{4}{2}\binom{2}{2}+$$
$$+\binom{5}{1}\binom{4}{1}\binom{3}{1}\binom{2}{2}+\binom{5}{1}\binom{4}{1}\binom{3}{1}\binom{2}{1}\binom{1}{1}=306$$

options to form metasymbols in a message when L=5.

In fact, this problem is related to other NP- complete combinatorial problems such as "The light bulb problem"[9], "The Problem of Context Sensitive String Matching"[10], "Low Autocorrelation Binary Sequences"[11] and the "Statistical mechanics and the partitions of numbers"[12].

In this work, we are searching metasymbols with no negative relative subindices and ordered from the lowest absolute position to the largest absolute position. This implies that the first symbol in a metasymbol has the lowest absolute position.

5. Recursive grouping

Once the metasymbols have been chosen, there exists the possibility that some metasymbols can be decomposed in more metasymbols over again. So, we may derive metasymbols from metasymbols. These we denoted as second order metasymbols. From these last we may derive third order metasymbols and so on.

Consider, for example, the message msg1 = "xyazbxyczd". We identify the next metasymbols

$$\alpha = a_2 * b_4 ** c_7 * d_9$$

$\beta = x_0 y_1 * z_3$

$|M| = 2$

$msg1 = \beta\alpha\beta$

The reconstruction of msg1 from the metasymbols is as follows:

$msg1 = \beta\alpha\beta$

$msg1 = xy*z$

$a*b**c*d$

$xy*z$

$msg1 = xyazbxyczd$

Now, we may examine another message msg2 = "xyazbxyczdxyazbxyczd" where we identify just one metasymbol

$\chi = x_0 y_1 a_2 z_3 b_4 x_5 y_6 c_7 z_8 d_9$

$|M| = 1$

It is clear that the metasymbol χ can be decomposed as metasymbols α y β, which become second order metasymbols. The recurrent decomposition for msg2 is:

msg2 = "xyazbxyczdxyazbxyczd"

$msg2 = \delta\delta$

where:

$\delta = x_0 y_1 a_2 z_3 b_4 x_5 y_6 c_7 z_8 d_9$

$\alpha = a_2 * b_4 ** c_7 * d_9$

$\beta = x_0 y_1 * z_3$

$\delta = \alpha\beta$

6. Experiments

We now briefly report on some experiments which were conducted with the purpose of determining experimentally if the ideas behind the metrics were effective.

6.1 Metrics evaluation through exhaustive search

We realized an exhaustive search with different strings trying to find the best repetitive patterns. Some results for metrics 1,2,3 and 4 are showed in table 4.

Table 4. Results of an exhaustive search evaluating metrics 1, 2, 3 and 4

Mensaje	F1	F2	F3	F4
xxxxy	M1 = x fm1=4 M2 =y fm2=1	M1 = xxx fm1=1 M2 =y	M1 = xxx Fm1=1 M2 =y	M1 = xxx Fm1=1 M2 = y
		fm2=1 M3 = x fm3=1 F=0.72	fm2=1 M3 =x fm3=1 F=0.78	fm1=1 M3 : x frec=1 F=0.78
xxxyy	M1 : yy frec=1 M2 : x frec=3 F= 0.81	M1 : xxy frec=1 M2 : x frec=1 M3 : y frec=1 F=0.78	M1 : xxy frec=1 M2 : x frec=1 M3 : y frec=1 F=0.78	M1 : xxy frec=1 M2 : x frec=1 M3 : y frec=1 F=0.78
Xyazxyb z	M1 : xyaxyb frec=1 M2 : z frec=2 F= 0.91	M1 : xyaz frec=1 M2 : x frec=1 M3 : y frec=1 M4 : z frec=1 M5 : b frec=1 F=0.55	M1 : xyaz frec=1 M2 : x frec=1 M3 : y frec=1 M4 : z frec=1 M5 : b frec=1 F=0.55	M1 : xyaz frec=1 M2 : x frec=1 M3 : y frec=1 M4 : z frec=1 M5 : b frec=1 F=0.55
abcdabcd	M1 : abcd frec=2 F= 0.0	M1 : abcd frec=1 M2 : a frec=1 M3 : b frec=1 M4 : d frec=1 M5 : c frec=1 F=0.55	M1 : abcd frec=1 M2 : a frec=1 M3 : b frec=1 M4 : d frec=1 M5 : c frec=1 F=0.55	M1 : abcd frec=1 M2 : a frec=1 M3 : b frec=1 M4 : d frec=1 M5 : c frec=1 F=0.55
zxxyyzz	M1 : xxyy frec=1 M2 : z frec=3	M1 : zx frec=1 M2 : yz frec=1 M3 : x frec=1 M4 : y frec=1 M5 : z frec=1	M1 : zx frec=1 M2 : yz frec=1 M3 : x frec=1 M4 : y frec=1 M5 : z frec=1	M1 : zx frec=1 M2 : yz frec=1 M3 : x frec=1 M4 : y frec=1 M5 : z frec=1

	F= 0.81	F=0.59	F=0.59	F=0.59

With the message "xxxxy" we have the next observations:

Metric 1 identifies that symbol x is repeated 4 times and that symbol y is different.

Metrics 2,3 y 4 build metasymbols unnecessarily.

With the message "xxxyy" we have the next observations:

Metric 1 identifies the fact that symbol x is repeated 3 times and that symbol y is repeated 2 times.

Metrics 2, 3 and 4 build metasymbols unnecessarily.

With the message "xyazxybz" we have the next observations:

Metric 1 does not identify that pattern xy*z is repeated twice, complemented by symbols a and b. In other words xyazxybz can be expressed thus:

xyazxybz=xy*z yxy*z
a b

Metrics 2,3 y 4 build metasymbols unnecessarily.

Exhaustive search requires extensive computational resources. In fact, those metrics have not been tested with strings of more than 12 characters. The time required on a PC @ 1GHz, 128MB-RAM to evaluate a string with 12 characters and just one metric was approximately 12 hours and we emphasize that demanded time growths exponentially.

As an alternative, we have programmed a Vasconcelos Genetic Algorithm [13] and we used metric 5 because this metric has shown the best result in all cases studied.

Now, we exemplify from the following experiment.

Message1 = "xyxxxwxx"
Vasconcelos Genetic Algorithm
Population size = 400
Number of generations = 100
Mutation probability Pm=0.85
Crossover probability Pc= 0.05
K=1.45
Length=8 characters
The best individual has:
F_5 = -0.366800
N = 3
M = 2
Groups:

y w length =2 repeated =0
1 5
x x x length =3 repeated =0
0 2 3
x x x length =3 repeated =1
4 6 7

Metasymbols:
$M=\{y_1w_5 , x_0x_2x_3 \}$
Time used = 11 seconds.

In figure 1 we show the metasymbols found in the message "xyxxxwxx"

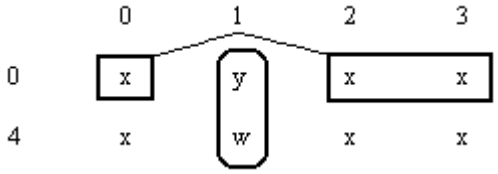

Figure 1. Metasymbols found in a message with 8 characters, $M=\{ y_1w_5 , x_0x_2x_3 \}$

Message2 = "xyAzxyBzxyCzxyDz"
Vasconcelos Genetic Algorithm
Population size = 400
Number of generations = 400
Mutation probability Pm=0.75
Crossover probability Pc =0.07
K=1.1
Length=16 characters
The best individual has:
F_5 = -0.012585
N = 5
M = 2
Groups:

A B C D length=4 repeated=0
2 6 10 14
x y z length =3 repeated =0
12 13 15
x y z length =3 repeated =1
4 5 7
x y z length =3 repeated =1
8 9 11
x y z length =3 repeated =1
0 1 3

Metasymbols:
$M=\{A_2B_6C_{10}D_{14} , x_{12}y_{13}z_{15} \}$
Time used = 2 minutes and 30 seconds.

In figure 2 we show the metasymbols found in the message "xyAzxyBzxyCzxyDz" using metric 5. The index in the message is calculated adding the horizontal and the vertical coordinate.

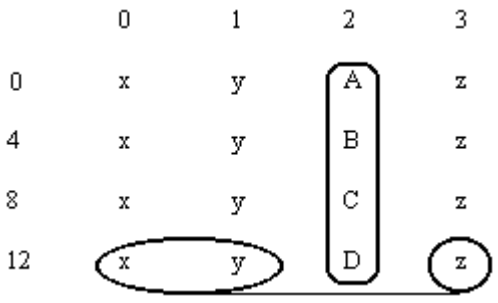

Figure 2. Metasymbols found in a message with 16 characters, M={$A_2B_6C_{10}D_{14}$, $x_{12}y_{13}z_{15}$ }

Message3 = "xrcaaxrrcacaxrcxaxrcxrcaaxrrcabc"
Vasconcelos Genetic Algorithm
Population size = 400
Number of generations = 400
Mutation probability Pm=0.75
Crossover probability Pc =0.07
K=1.3
Length =32 characters
The best individual has:
F_5= -0.000089
N = 9
M = 5
Groups:

x a x x 5 11 20 25	length =4 repeated =0
r c a b 13 14 24 30	length =4 repeated =0
x a r c 12 16 18 31	length =4 repeated =0
r c a 7 8 9	length =3 repeated =0
c x x r 10 15 17 26	length =4 repeated =0
x a r c 0 4 6 19	length =4 repeated =1
r c a 21 22 23	length =3 repeated =1
r c a 27 28 29	length =3 repeated =1
r c a 1 2 3	length =3 repeated =1

Metasymbols:
M={$x_5a_{11}x_{20}x_{25}$, $r_{13}c_{14}a_{24}b_{30}$, $x_{12}a_{16}r_{18}c_{31}$, $r_7c_8a_9$, $c_{10}x_{15}x_{17}r_{26}$ }
Time used = 8 minutes and 5 seconds.
In figure 3 we show the metasymbols found in the message "xrcaaxrrcacaxrcxaxrcxrcaaxrrcabc"

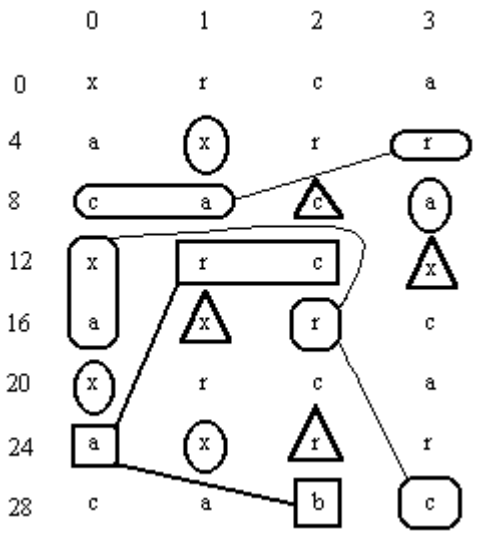

Figure 3. Metasymbols found in a message with 32 characters, M={$x_5a_{11}x_{20}x_{25}$, $r_{13}c_{14}a_{24}b_{30}$, $x_{12}a_{16}r_{18}c_{31}$, $r_7c_8a_9$, $c_{10}x_{15}x_{17}r_{26}$}

Case 1. The metric found in 11 seconds that the metasymbol $x_0x_2x_3$ was repeated twice and that metasymbol y_1w_5 was the complement of the message. In figure 1 we can easily identify the three groups in which the message "xyxxxwxx" was divided.
In case 2, the metric is able to find that a pattern with no contiguous symbols x, y and z is repeated three times which is very good. Besides, a minimal of two metasymbols were found and the length of each one is not slant to the extreme values 1 and 16. The time required for this result was approximately 2 minutes which compared with the exhaustive search is too many times lower. In figure 2 we can easily see how the metasymbol $x_{12}y_{13}z_{15}$ is repeated four times.
In case 3, the metric found a set of 5 metasymbols from a partition of 9 groups, and that is very good. In figure 3 metasymbols can be easily identified by means of triangles, squares, circles, rectangles and ellipsoids. Groups weren't drawn because they make less clear the visualization. As we can see, metasymbols has no contiguous symbols and theirs lengths aren't slant to the extreme values 1 and 32. The time required for this result was approximately 8 minutes which is a very good in comparison with the time required by the exhaustive search.
So we can say, metric 5 has provided the best result in comparison to the other metrics (1,2,3 and 4). Metrics 1,2,3 and 4 fails with simple strings while metric 5 has been probed successfully with strings of 16 and 32 characters. Here we have shown a few results in order to illustrate its performance.

7. Conclusions

Exhaustive search of metasymbols is not to be recommended because of its computational complexity. As an alternative to reduce the search time is possible to use a genetic Algorithm and in particular a Vasconcelos Genetic Algorithm where the measure of fitness can be metric 5. In order to code a message decomposed in metasymbols and compare the result with the original Huffman coding we intend to test with messages of larger length.

It is very important to stress that the problem we are attempting to solve has applications in various fields. For example, it may be applied in cryptography, large data bases matching, internet search engines, automatic translation, etc.

The analysis of exploration of the metrics mentioned above does not convey the inherent difficulty which is found in the problem. Furthermore, we intend to approach the problem with the aid of alternative soft computing techniques, such as multilayer backpropagation networks and support vector machines. These tools will enable us to attempt a classification of the groups of metasymbols to make our search more efficient.

8. References

[1] Pierce, J. R., *An Introduction to Information Theory*, 2nd Ed., Dover, 1980, 94-101.

[2] Shannon, C.E., *A Mathematical Theory of Communication*, Bell Sys. Tech. *J. 27* (1948), 379-423, 623-656.

[3] Hamming, R.W., *Coding and Information Theory*, Prentice-Hall, 1980, p. 80-89.

[4] Shannon, C., op. cit., p. 7.

[5] Burrows M., and Wheeler, D. J., *A block-sorting lossless data compression algorithm,* Digital Syst. Res. Ctr., Palo Alto, CA, Tech. Rep. SRC 124, May 1994

[6] J.L. Bentley, D.D. Sleator, R.E. Tarjan, and V.K. Wei. *A locally adaptive data compression algorithm*, Communications of the ACM, Vol. 29, No. 4, April 1986, pp. 320–330

[7] Huffman, D. A.. *A method for the construction of minimum-redundancy codes*. *Proc. Inst. Radio Eng. 4*0, 9 (.), 1098–1101, Sept 1952.

[8] Nelson Mark, Jean Loup Gailly, *The Data Compression Book*, Second Edition, M&T Books Redwood City, CA (1995).

[9] Paturi, R., Rajasekaran, S., and Reif, J.H. (1989), *The Light Bulb Problem.* In Second Work shop on Computational Learning Theory.

[10] Venkatesan T. Chakaravarthy and Rajasekar Krishnamurthy, *The Problem of Context Sensitive String Matching,* Computer Science Department, University of Wisconsin Madison, WI 53706, USA.

[11] Steven Prestwich, *A Hybrid Local Search Algorithm for Low Autocorrelation Binary Sequences,* Technical Report, Department of Computer Science, National University of Ireland at Cork.

[12] F.C. Auluck and D.S. Kothari, *Statistical mechanics and the partitions of numbers*, Proceedings of the Cambridge Philosophical Society 42 (1946).

[13] Kuri, A., *A Comprehensive Approach to Genetic Algorithms in Optimization and Learning,* Editorial Politécnico, 1999.