International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Transforms Based on Ignorance (TBIs): Their Mode of Operation and Analysis Complexity

Abderrahim SABOUR

Ibn Zohr University, Higher School of Technology, Department of Computer Science, B.P: 33/S Agadir, Morocco

Abstract: This paper introduces the Transforms Based on Ignorance (TBIs) through the presentation of functions and their classes in three states. The objective is to evaluate the complexity introduced by the concept of ignorance for proposing a new model based on the analysis of the Graphs of Perceived Neighborhood Distributions (GPND). The purpose of this study is to present TBIs, analyze their properties, and evaluate them in order to find new applications.

Keywords: TBI; GPND; three states function; complexity

1. Introduction

To allow readers to better understand the problem treated, it is necessary to raise the following questions:

Q1- what is the point of providing a population the right to lie?

A- To simulate the evolution of a population where distrust governs by requiring all individuals to lie permanently on their identity, presents no interest at first sight. But an original application will exploit the emerging complexity to create a new class of binary sequences regenerators cryptographically secured and based on a new concept called TBIs.

Q2- how TBIs will allow an individual to lie?

A- A transformed based on ignorance can be perceived as a mask that the individual use whenever they interact with their environment and will be reset after each use. In other words, in this approach, any interaction of a given individual with his environment involves passing by a TBI which the parameters are calculated after each use depending on the system state.

To be interested in the study of ignorance and transformed based on this concept is justified in two ways: the first theoretical is to explore every possible source of complexity and try to quantify it. The other axe would be to find its practical applications.
Our work is the first to take an interest in ignorance as a source of complexity. We will try to formalize this concept to quantify the emerging complexity of its implementation through the introduction of a model based on the analysis of distribution’s graphs of neighborhoods. This introduces new questions and a new approach given that this research of transformed based on ignorance is still at the beginning. So in what follows we will present the transformed based on ignorance TBI followed by the proposed model to formalize this concept and an introduction of the concept of neighborhood and graphs of neighborhood’s distributions whose operation requires the introduction of a model based on the juxtaposition of graphs. This approach offers the possibility to make a first estimate of the complexity because it introduces a simplification of the problem.

First, these transformed based ignorance have been introduced and used to create a binary suites regenerator crypto-graphically safe named 'RA NMJ' [1] [2] which ranks as a behavioral evolutionary algorithm simulating the chaos [7] [8] [9] created by a population of individuals who are forced to operate in a universe where distrust is dominating and which requires for each individual and in every moment to lie about his identity.

To make an artificial population of individual, created from a password of any size, who can keep their identities undetectable, necessitated the search for original source of complexity, flexible and easily adaptable.

The evolution's analyses of the populations of individuals adopting behaviors based on mistrust are characterized by:

- Lack of fitness function;
- An individual's age is a discriminating factor (older are massacred first).
- A mutual distrust between individuals in the population: the main concern of each individual is to present a false identity for each interaction.

2. The R.A NMJ algorithm and the transformed based on ignorance

The RA NMJ [1] algorithm is a regenerator of binary sequences that are cryptographically secured based on evolutionary methods. It simulates a dissipative nonlinear dynamical system with compensation based on the transformed based on ignorance. It proceeds in three phases: after creating the initial population using a class of three states function on a password of any size, the result of this first step is the creation of a population of individuals associated to this password while ensuring a high sensitivity to initial conditions (states of the bits of the password and its length).The second process is iterative. It allows the initial population to become more mature through a set of coupling function and order function. The result is a population of individuals whose number can be equal to or different from number of individuals at the start of the process but with Block Data sizes multiplied by the number of iterations of the process II. The process III iteratively allows evolving the population while adding a contribution function where each individual is required to regenerate a binary sequence. These binary sequences are concatenated to be xored with the clear message.

3. Notations and Definitions

Definition 1:
The TBIs are defined as functions with multi-state and at least one state initiating the process of ignorance.
This concept has already been introduced by the three state functions (α, β and λ) [3][4] and whose ignorance is associated with the β state. In all what follows, the study will focus on the TBIs defined by functions in three states since it represents the type of TBIs used by the RA NMJ algorithm.

Definition 2:
We call a three states function α, β and λ, every function of ℕ in{α, β, λ}.

Definition 3:
In each function of two or three states **F**, we associate a unique sequence **f** defined by: **f = F(0)F(1)...F(n)**... And if there is an integer k such that **f = F(0)F(1)...F(k)F(0)F(1)...F(k)...**, we say that **F** is periodic with period **F(0)F(1)...F(k)**, and more if k is the smallest integer, **F(0)F(1)...F(k)** is called primitive signal of **f**. We denote by F. Subsequently, in this case, $F(n) = F(n \,\%\, L(Sp(F)))$ for each n∈N.

If **f** is a finite sequence, it is extended to a single periodic infinite sequence whose length of the original signal is a divisor of **f**'s length, Called regenerative signal **F**. We denote by $S_R(\mathrm{F})$ a concatenation of the original signal.

4. The Functioning of the Transformed Based on Ignorance TBI

This section presents the algorithm for calculating the state of the k^th bit of S according to the suite E and the observer G without calculating k-1 previous values.

TBI is defined by the observer G = [x, y, z, t] where: x is the number of occurrences of the state α, y and t the number of occurrences of β and z is the number of occurrences of the state λ. Let the observer G = [x, y, z, t] and the associated values of DX, DY, DZ and DT authenticates the set of observers involved.

The original version of R.A NMJ uses a class of three states functions defined by:

DX= |2:7|, DY=|1:6|, DZ=|2:7|, DT= |1:5|
To calculate S (k) as a function of E (k) and G

The algorithm of calculating the state of the k^th bit of the vector Si, knowing the observer Gi.

Algorithm 1 : calculation of S(k)

Input: G : observer defined by [x, y, z, t]
E : binary sequence of length lg (E)
k : the index of the position
Output: Stat : the state of the bit at position k of S
1 posi ← k + (y + t). (k/(x + z))
2 dec ← (k%(x + z))
3 if dec ≥ x then
3-1 Stat ← not(E((posi + y)%lg(E)))
4 else
4-1 Stat ← E((posi)%lg(E))
5 end if

This code is used to calculate the value of the bit at position k of the S and this by applying an observer G defined by x, y, z, and t. With x is the number of occurrences of the state α, y and t the number of occurrences of the state β and z number of occurrences of the state λ. The variable pos i allows the calculation of the bit position corresponding to the value k, the calculation of the variable dec indicates the status applied: if the value of dec ≥ x the observer applies the state λ, otherwise it applies the state α.

Example 1: Let the bit string E composed of 9 bits labeled by numbers from 1 to 9, indicating their positions.

E :	1	2	3	4	5	6	7	8	9

By applying the observer G1, defined by G1 = [2,2,2,2], on the input E and setting as length of the vector S resulting in the value 11, we have:

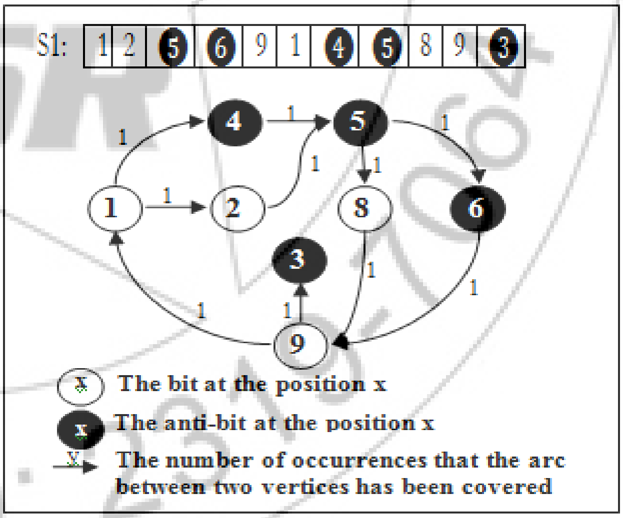

Figure 1: Neighbourhood distribution created by the application of the observer G1 on the signal E

The application of an observer G on input E can be seen as creating a new **distribution between neighboring** elements composing the input E, and this by breaking some existing neighborhood links and are substituted by the definition of fictitious temporary bonds due to the application of the states β and λ. The influence of the stateλ will be marked by the introduction of anti-bite notion and it leads to the association of two nodes to each bit of E.

Example 2: By applying the observer G2, G2 defined by = [2,3,2,3], on the input E while keeping the same length vector S1 to S2, then:

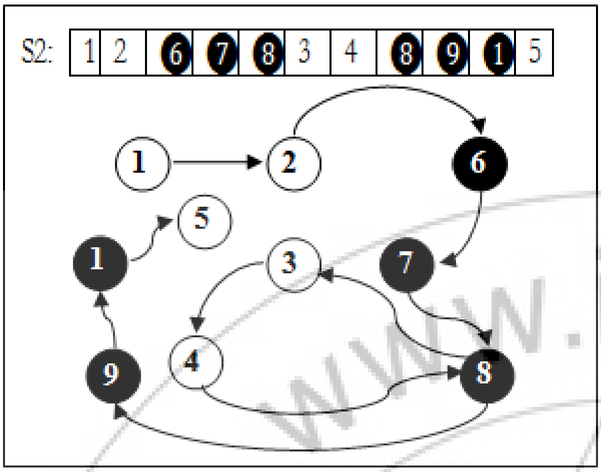

Figure 2: Neighbourhood distribution created by the application of the observer G2 on the signal E

Example 3: By applying the observer G3, G3 defined by = [2,6,2,9], on the E input while keeping the same length of S1 to S3 vector, we have:

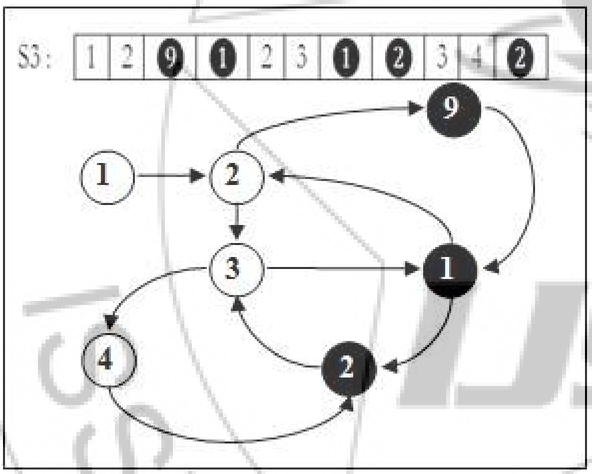

Figure 3: Neighbourhood distribution created by the application of the observer G3 on the signal E

Notes:
The algorithm presents several shortcomings:

- The treatment always begins with the bit index 1 which reduces the space of existence of the vectors Si.
- The introduction of a browse direction will double the existing space of the vectors Si.
- The graph G3: $(1,\bar{1},2,\bar{2},3,4,\bar{9})$ shows the coexistence of bits and anti-bits associated to the bits with the indices 1 and 2.

How to interpret this coexistence?
- The two graphs G2: $(1,2,3,4,\bar{6},\bar{7},\bar{8})$ and G3: $(1,\bar{1},2,\bar{2},3,4,\bar{9})$ are isomorphic since the renaming of vertices: $\bar{6}$ by $\bar{9}$, $\bar{7}$ by $\bar{1}$ and $\bar{8}$ by $\bar{2}$ will allow the passage of the graph G2 to G3.

The algorithm of calculating the state of the kth bit of the vector Si, knowing the observer Gi and taking into consideration the position and direction.

Algorithm 2 : calculation of $S(k)$ considering ps, ss

Input : G : observer defined by $[x, y, z, t, ps, ss]$
E : binary sequence of length lg (E)
k : the index of the position
Output: Stat : the state of the bit at position k of S
1 posi $\leftarrow k + (y + t).(k/(x + z))$
2 dec $\leftarrow (k\%(x + z))$
3 if ss $\neq$ 0 then
3.1 posi $\leftarrow$ ps + posi
3.2 if dec $\geq$ x then
3.2.1 $S(k) \leftarrow$ not$(E((posi + y)\%lg(E)))$
3.3 else
3.3.1 $S(k) \leftarrow E((posi)\%lg(E))$
3.4 end if
4 else
4.1 posi $\leftarrow$ ps $-$ posi
4.2 if posi $<$ 0 *then*
4.2.1 posi $\leftarrow lg(E) - (-posi)\%lg(E)$
4.3 end if
4.4 if dec $\geq$ x then
4.4.1 $S(k) \leftarrow$ not$(E((posi + y)\%lg(E)))$
4.5 else
4.5.1 $S(k) \leftarrow E((posi)\%lg(E))$
4.6 end if
5 end if

The algorithm of calculating the state of the k^{th} bit of the vector Si, knowing the observer Gi and taking into account the position and direction.

Example 4: Let the bit string E

E :	1	2	3	4	5	6	7	8	9

By adding to the observer G1 '= [x, y, z, t, p, ss] the two fields ps which means the position and ss field which means the direction of the route since the vector can be traversed in both directions . For example let G1' = [2,2,2,2,77,1], applied to the same input E and for an output S1' having a length equal to 11. The position 77 is not defined because the length lg(E) = 9. In order to get around this problem the new position defined by ps←ps % lg(E). By convention the ss field code two states that indicate the direction of the route to apply the value 0 indicates the direction of the route from left to right and the value 1 indicates the opposite direction and we have:

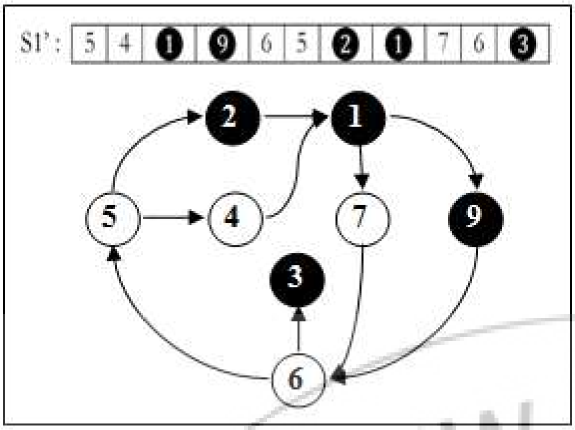

Figure 4: Neighbourhood distribution created by the application of the observer G1' on the signal E

Example 5: Let G2 is defined by $G2' = [2,3,2,3,115,1]$ we have:

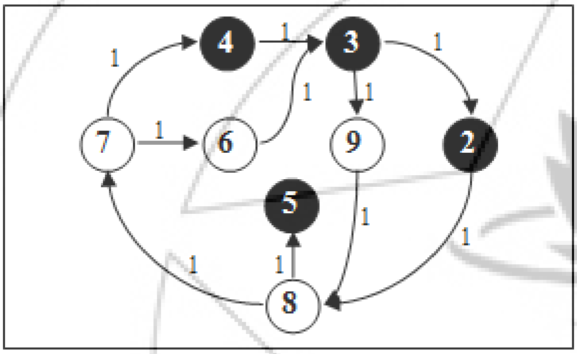

Figure 6: la distribution de voisinage créer de l'application de G2' sur le signal E

To justify the use of position field "ps" and the direction field "ss". One can quickly see that in the case where the position and direction are defined as bits S [1] ... S [Min (Ax)] are identical and this whatever the observer applied. Free information which greatly reduces the search space simplifying the prediction of vector E. The representation as a graph of the perception Si of an observer Gi of the E reality shows in a clear way the new neighborhood distributions that will be used to study the emerging complexity of the processed based on ignorance which is the mechanism adopted by the RA NMJ algorithm in order to equip the population of individuals with the faculty to lie by ensuring that the process of lying should be easily calculated and allow regeneration of many lies.

The issues raised by TBIs are:

- How to estimate their complexity?
- How can we compare two classes of function based on ignorance by defining a measure to quantify?

The following paragraph introduces the distribution's graphs of neighborhood perceived GPNDs and how their juxtaposition allows estimating the complexity associated with all TBIs in a given class.

5. The Graphs of Perceived Neighborhood Distributions (GPND)

The foregoing allowed associating with each TBI a path, the algorithm 3 associates to each path the corresponding GPND. The definition of GPNDs comes to circumvent the problem introduced by the juxtaposition of the connection between the summits of resulting graph.

The advantage of this representation is that it facilitates the process of juxtaposition of p-graphs as connectivity between two vertices Si and Sj, is measured by the existence of an arc between two vertices and not the existence of a path, hence the designation of a new term that is GPND for the Distribution Graph of the Perceived Neighborhood: in GPND the existence of a path between two vertices does not imply the connectivity between two vertices, only the existence of an arc that imply connectivity. This choice was necessary to prepare the p-graphs associated with bag paths processes juxtaposition.

3.1 Creation Process of GPND

The construction of the graph G1 is made in an incremental manner. and this by traversing the path from the start vertex to the top of arrival and choosing at each iteration the current vertex as elected and inserting arcs valued in the sources correspond to the vertices already visited and having as destination the chosen vertex. The value of the arc is equal to the minimum length of the path between the top address and the source vertex of occurrences.

Algorithm 3: calculating p-graph G associated with a path Ch

Input: Ch[N]: an array of N elements indicating the order of path vertices.
Output : G : partial p-graph oriented value
S: list structure used to label the vertices already visited.

```
1 S.add(Ch[1],1)
2 for i in 2 .. N
2.1 for each visited_S in S
2.1.1 G.arcAdd(visited_S, Ch[i], minDist(visited_S, Ch[i]))
2.2 end
2.3 S.add(Ch[i],i)
3 end
```

Note:

- The S.add function adds the summit Ch [i] at position i to the list of visible summits S. It takes two arguments: the first is the number of the summit; the second is its position. If the top is not labeled then it will be added to the list with the current position else the function updates the value of its position.
- The processing of path translates into the browse of that first peak to the last. And the processing of the current summit will result in the definition of a set of arcs whose valued sources correspond to visited vertices and which is having as destination the current vertex, each arc will get

as value the length of the minimal path between the two vertices.

3.2 Applications

Let two paths C1 and C2 defined by:

C1: a – b – c – a – b – c – a – b – c
C2: a – b – c – c – b – b – b – a – c

We seek to determine the graphs of neighborhood's distribution perceived G1 and G2 respectively associated with paths C1 and C2

Example 6: Let the path:
C1: a – b – c – a – b – c – a – b – c

The application of Algorithm 3 gives:

a: S.add(a)
a – **b** : G1.acrAdd(a,b,minDist(a,b)) ; S.add(b) avec minDist(a,b)=1
a – b – **c** : G1.acrAdd(a,c,2) ; G1.acrAdd(b,c,1) ; S.add(c)
a – b – c – **a** : G1.acrAdd(a,a,3) ; G1.acrAdd(b,a,2) ; G1.acrAdd(c,a,1) ; S.add(a)
a – b – c – a – **b** : G1.acrAdd(a,b,1) ; G1.acrAdd(b,b,3) ; G1.acrAdd(c,b,2) ; S.add(b)
a – b – c – a – b – **c** : G1.acrAdd(a,c,2) ; G1.acrAdd(b,c,1) ; G1.acrAdd(c,c,3) ; S.add(c)
a – b – c – a – b – c – **a** : G1.acrAdd(a,a,3) ; G1.acrAdd(b,a,2) ; G1.acrAdd(c,a,1) ; S.add(a)
a – b – c – a – b – c – a – **b** : G1.acrAdd(a,b,1) ; G1.acrAdd(b,b,3) ; G1.acrAdd(c,b,2) ; S.add(b)
a – b – c – a – b – c – a – b - **c** : G1.acrAdd(a,c,2) ; G1.acrAdd(b,c,1) ; G1.acrAdd(c,c,3) ; S.add(c)

Thus the graph G1 will look as follows, with:

- The vertices correspond to the covered vertices.
- The arches are labels in the format (Say: Occ) with Dis is the length of the path between the departure node and the destination node. Occ is the number of times that this was perceived

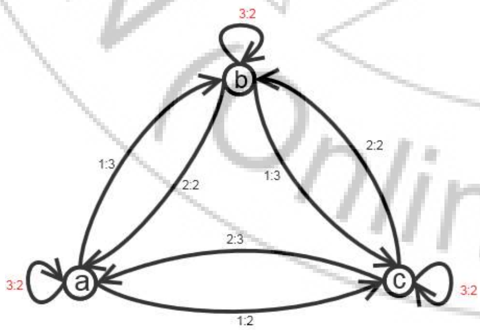

Figure 2 : GDVP G2 associated to the path C2

Example 7: Let the path:

C2: a – b – c – c – b – b – b – a – c

The application of algorithm 3 gives:

a : S.add(a)
a – **b** : G2.acrAdd(a,b,1) ; S.add(b)
a – b – **c** : G2.acrAdd(a,c,2) ; G2.acrAdd(b,c,1) ; S.add(c)
a – b – c – **c** : G2.acrAdd(a,c,3) ; G2.acrAdd(b,c,2) ; G2.acrAdd(c,c,1) ; S.add(c)
a – b – c – c – **b** : G2.acrAdd(a,b,4) ; G2.acrAdd(b,b,3) ; G2.acrAdd(c,b,1) ; S.add(b)
a – b – c – c – b – **b** : G2.acrAdd(a,b,5) ; G2.acrAdd(b,b,2) ; G2.acrAdd(c,b,2) ; S.add(b)
a – b – c – c – b – b – **b** : G2.acrAdd(a,b,6) ; G2.acrAdd(b,b,1) ; G2.acrAdd(c,b,3) ; S.add(b)
a – b – c – c – b – b – b – **a** : G2.acrAdd(a,a,7) ; G2.acrAdd(b,a,1) ; G2.acrAdd(c,a,4) ; S.add(a)
a – b – c – c – b – b – b – a – **c** : G2.acrAdd(a,c,1) ; G2.acrAdd(b,c,2) ; G2.acrAdd(c,c,5) ; S.add(c)

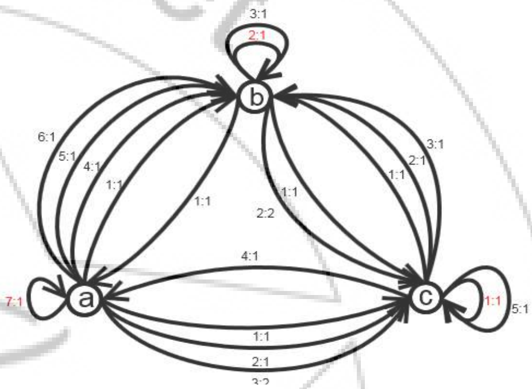

Figure 3 : GDVP G2 associated to the path C2

G2 is a partial p-value graph where each arc shows the length of a path and the number of occurrences that this measure was collected.

3.3 Juxtaposition Process

Definition 4:

A graph of Perceived Neighborhood Distributions GPND is a p-graph G (S, A, ν) with:

S: the set of summits
A: the set of arcs or edges
ν: the application of the valuation of the graph associates to each arc (or edge) a couple (Dis: Occ) where Dis is the path length, and Occ represents the number of times that the length Dis was measured.

Definition 5:

The juxtaposition of a set of GPNDs$\{G_1(S_1, A_1, \nu_1), \ldots, G_n(S_n, A_n, \nu_n)\}$will regenerate an overall p-graph$G_g(S_g, A_g, \nu_g)$avec with : $S_g = \bigcup_{i=1}^{n} S_n$, $A_g = \bigcup_{i=1}^{n} A_n$ with for each subset of arcs (or edges) having the same path length, the same starting vertex and the same finish top, this subset will be replaced by a single arc (or edge) having the same length, the same starting vertex and the same finish top but the number of occurrences is equal to the sum of numbers of occurrences of all the arcs of this subset.

The juxtaposition of p-graphs to create a global graph associated with a bag of path itself associated with a class of three-state function, since it comprises for each function three-stateway. The analysis of the overall graph will assess the collective influence of the class in question.

The following figure presents the overall graph resulting from the juxtaposition of two GPNDs G1 and G2.

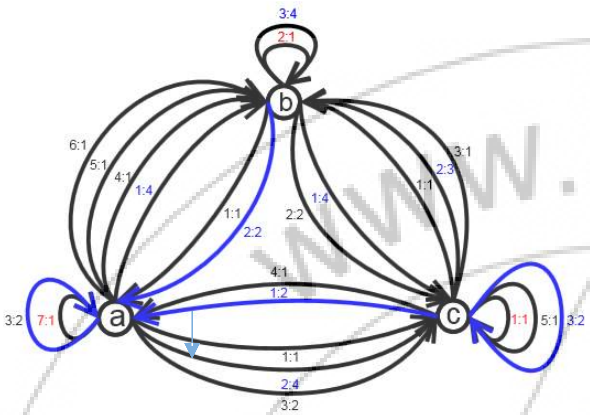

Figure 4 : the overall graph resulting from the juxtaposition G1 and G2

The analysis of the associated graphs to the perceived neighborhood's distributions, resulting from the application of a set of TBI class defined by a tri state function, follow the following steps:

1) Juxtaposition: as well the juxtaposition allows combining in single p-graph information on the neighborhood's distributions received by each TBI in order to estimate the collective complexity.
2) Reduction and standardization matrix to extract a distance between the vertices of the graph starting with calculating a distance between the two vertices to two. Standardization consists in canceling the values of the diagonal and make symmetric matrix.
3) MDS Application: The application of MDS Multidimensional scaling [5][6] is used to infer the dimension of the Euclidean space that can meet the constraints of distances associated with distributions of perceived neighborhood.

Note:

Consider a set of n points satisfying the following constraint: Whatever two different points ni and nj of this ensemble, there is always the distance between two points is measured and equal to a nonzero constant. The fact that the distance is measured, and the same whatever two points implies that the size of the minimum Euclid space that can satisfy these constraints distance is equal to the number of the set point - 1.

Proof 1: trivial by induction, trying to put n points in a Euclidean space in an incremental way, we begin by positioning 2 points, 3 points to the nth point, knowing that the minimum dimension of the Euclidean space, which may contain any point equal to 0.

Definition 6:
The **occupancy rate** is defined as the ratio between the minimum dimension of the Euclidean space that can satisfy the constraints of distance between all pairs of points and the maximum dimension.

6. The estimation of the complexity associated to the classes of tri state functions

In this section we try to exploit the fact that the first process of the algorithm RA NMJ dedicated to the creation of the initial population, have a special feature: the distribution of neighborhood created does not depend on the contents of bits passwords, but depends on their number and characteristics of the function class tri state applied over the length of the regenerated channels. Faced with this situation the idea of studying the behavior of function classes based on ignorance (TBIs) through the analysis of associated paths sacks was adopted. Thus initially the idea was to study the characteristics of each path, when applying a transform based on ignorance TBI associated with this case has a function in three states. Support of the state λ is manifested by the association to every bit of two summits; the overall graph has as number of top, the number of bits of the treated chain doubled. Each bit has two summits: one corresponding to it when it is reachable via the α state and another summit when the bit is reachable via the state λ.

This section uses the model developed to compare the tri state function class implemented by the RA NMJ algorithm (TH class) with three variants of the new class named: **C1**, **$C1_{STR}$** and **$C1_{SauT}$**.

Let the observer G = [x, y, z, t, p, ss], the ranges of values, denoted Di, which may take x, y, z and t define the tri-state function class. The original version of the algorithm RA NMJ uses during the creation of the initial population, the class of functions defined by three states:

Class **TH**: $D_x = |2:7|$, $D_y = |1:6|$, $D_z = |2:7|$ and $D_t = |1:5|$.
and we define another class named C1 with:
Class **C1:** $D_x = |2:5|$, $D_y = |1:8|$, $D_z = |2:5|$ and $D_t = |1:7|$.
Along with two variables of class C1
Class **$C1_{STR}$**: $D_x = |2:5|$, $D_y = |1 + \mathrm{STR}: 8 + \mathrm{STR}|$, $D_z = |2:5|$ and $D_t = |1 + \mathrm{STR}: 7 + \mathrm{STR}|$.
With $\mathrm{STR} = \frac{lg\ (E)}{64}$, with $lg\ (E)$ is the number of bits of the sequence E.
For the three classes **TH, C1** and **$C1_{STR}$**
ps_k: denotes the index of the start bit of the application of the k^{th} TBI. It takes the values of $|0{:}\lg(E) - 1|$, initialized to 0:

$$\begin{cases} ps_0 = 0\ si\ k = 0 \\ ps_{k+1} = [ps_k + (lg(E) * 17) mod(x * y + z * t)]\ mod\ lg(E)\ k \geq 1 \end{cases}$$

Algorithm 4 : value of the variable 'pas' without skip

Input : G : observer defined by $[x, y, z, t]$
$posi$: indicate the current state of the observer G
Output : pas : the number of bits that the observer must skip
1 $if\ posi\ <\ x\ then$
1.1 $Sig \leftarrow 1; pas\ \leftarrow 1; posi \leftarrow posi + 1;$
2 $else$
2.1 $if\ posi =\ x\ then$
2.1.1 $Sig \leftarrow 0; pas\ \leftarrow y; posi \leftarrow posi + y;$
2.2 else
2.2.1 $if\ posi <\ x + y + z\ then$
2.2.1.1 $Sig \leftarrow 0; pas\ \leftarrow 1; posi \leftarrow posi + 1;$
2.2.2 $else$
2.2.2.1 $Sig \leftarrow 1; pas\ \leftarrow t; posi \leftarrow 0;$
2.2.3 $end\ if$
2.3 $end\ if$
3 $end\ if$

Finally, the $C1_{SauT}$ class defined by: $D_x = |2{:}5|, D_y = |1{:}8|, D_z = |2{:}5|\ and\ D_t = |1{:}7|$.

The class $\mathbf{C1_{SauT}}$ uses a skip function triggered by the transition from the state α to the state β once each three cycles. Normally the transition from the state α to the state β is translated by a skip with a constant value equal to the value ofyassociated to the current TBI, the class$\mathbf{C1_{SauT}}$ introduce a recurrent sequence U_n defined by:

$$\begin{cases} U_0 = x * z + y * t \\ U_n = (2 * U_{n-1} + 1) mod\ 1000 \end{cases}$$

The major difference of the $\mathbf{C1_{SauT}}$ class relating to classes **TH**, **C1**, $\mathbf{C1_{STR}}$ is that $\mathbf{C1_{SauT}}$ introduces steps whose value varied according to the following sequence $U_n$once every several cycles, in this case the number of cycles is set to 3.
The values taken by the variable 'pas' by executing the algorithm 4 are: 1, y and t. the algorithm 5 add to these values the expression y +$U_n$and whose the value changes.
The qualities of a step function:
1- Take in charge when it is possible, of the chain lengths that will be treated and regenerated.
2- Take charge the parameters of the applied TBI
3- Be simple to calculate: reduced complexity
4- Have a maximum period
5- Regenerate global graphs with maximum connectivity within the meaning of GPNDs

Algorithm 5 : value of the variable 'pas' with skip

Input : G : observer defined by $[x, y, z, t]$
$posi$: indicate the current state of the observer G
Output : pas : the number of bits that the observer must skip
1 $if\ posi\ <\ x\ then$
1.1 $Sig \leftarrow 1; pas\ \leftarrow 1; posi \leftarrow posi + 1;$
2 $else$
2.1 $if\ posi =\ x\ then$
2.1.1 $Sig \leftarrow 0; pas\ \leftarrow y; posi \leftarrow posi + y;$
2.1.2 $if\ flip =\ true\ then$
2.1.2.1 $pas\ \leftarrow pas + Un; flip \leftarrow false;$
2.1.3 end if
2.1.4 $occ\ \leftarrow occ + 1$
2.2 else
2.2.1 $if\ posi <\ x + y + z\ then$
2.2.1.1 $\mathrm{Sig} \leftarrow 0; \mathrm{pas}\ \leftarrow 1; \mathrm{posi} \leftarrow \mathrm{posi} + 1;$
2.2.1.2 $\mathrm{Un} \leftarrow (2 * \mathrm{Uo} + 1) \mathrm{mod}\ 1000$
2.2.1.3 $\mathrm{Uo} \leftarrow \mathrm{Un}$
2.2.2 else
2.2.2.1 $\mathrm{Sig} \leftarrow 1; \mathrm{pas}\ \leftarrow \mathrm{t}; \mathrm{posi} \leftarrow 0;$
2.2.2.2 if $\mathrm{occ} = 2$ then
2.2.2.2.1 $\mathrm{occ}\ \leftarrow 0; \mathrm{flip} \leftarrow \mathrm{true};$
2.2.2.3 end if
2.2.3 end if
2.3 end if
3 end if

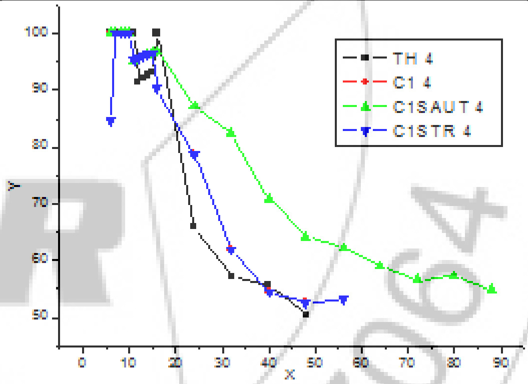

Figure 5 : Lg(S)=4, with support of the state λ

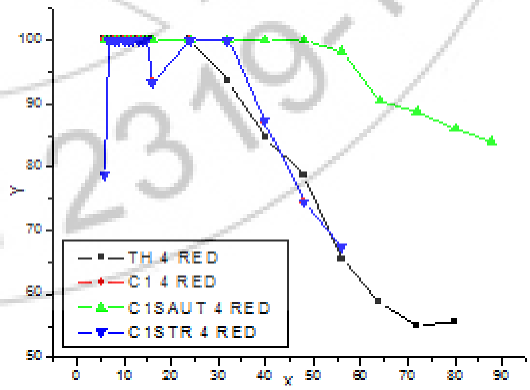

Figure 6 : Lg(S)=4, without support of the state λ

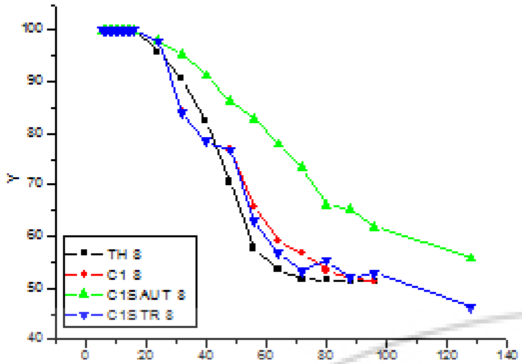

Figure 7 : Lg(S)=8, with support of the state λ

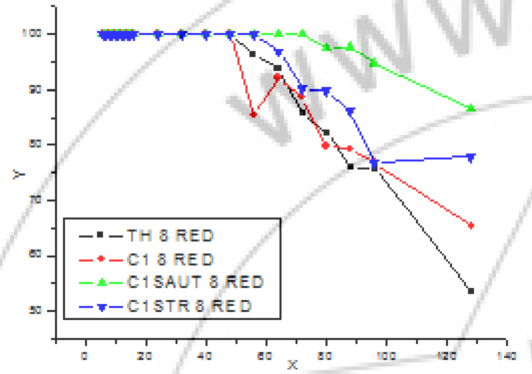

Figure 8 : Lg(S)=8, without support of the state λ

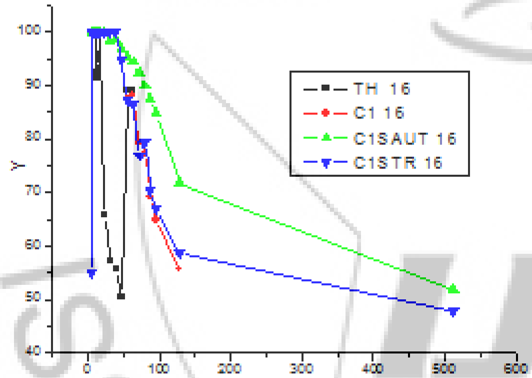

Figure 9 : Lg(S)=16, with support of the state λ

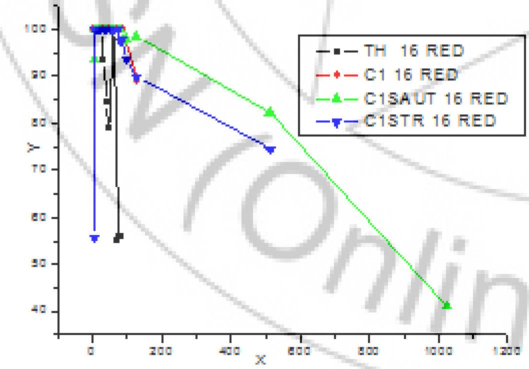

Figure 10 : Lg(S)=16, without support of the state λ

The X axis represents the number of summit of the graph and The Y axis represents the ratio between the minimum dimensions. Euclidean space that can satisfy constraints neighborhood and the upper bound.

7. Conclusions and perspectives

This paper has formally introduced the TBIs and the estimation process of their complexity as:

a) It has proposed an algorithm which allows the direct calculation of the value $S_G(i)$, which makes the realization of circuit that implements this concept based on mere ignorance as it is introduced by the TBIs.
b) It has calculated the emerging complexity of TBIs: due to the difficulties encountered when analyzing sacks of the paths browsed (SCP) and despite the power of this model, this approach was partially abandoned in favor of the analysis of the neighborhood distributions. This exploitation of the concept of neighborhood has allowed the development of a gateway to a powerful mathematical formalism, which is the graph theory, since it associates each TBI, a partial weighted and directed p-graph.
c) It has explored partial weighted and directed p-graphs associated with each TBI required the use of successive simplifications as well the juxtaposition process of all the p-graphs associated to a TBI class, regenerates an overall weighted and oriented graph where the summits represent the bits states of the input E and the list of arcs between two summits data indicate the lengths of the perceived paths between these two summits and the number of occurrences that each path length was perceived.

The fact that the measurements performed on binary sequences of higher lengths, give p-graphs where even the juxtaposition is unable to allocate paths between the different graph summits. Despite this limitation, this model allowed to compare and justify the choice of a class compared to others, it also shows the added value of the skip function.

As prospect the research for models allowing the estimation of the complexity of a TBI without appeal for an estimation of complexity partners of a class of TBIs and without care of the length of the entry or that of the release. Besides, they can be used to find other scopes of these TBIs.

8. Acknowledgment

The Author Would Like To Thank The Following People For Their Invaluable Contributions:
The late Professor Aboubakr Lbekkouri for his enthusiastic guidance, his advices and suggestions.
Dr. Ahmed Jadir, Professor Mohamed Akbour and Anass El ghazi for their suggestions in editing this paper

References

[1] A. Sabour, A. Asimi and A. Lbekkouri, Genetic Regenerator of Pseudo-random Sequences R.A NMJ, IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.1, January 2007
[2] A. Sabour, The Study of Sensitivity of R.A NMJ Algorithm, IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.1, January 2012
[3] A. Sabour, A. Asimi and A. Lbekkouri, The Three States Functions: Theoretical Foundations and Estimated

Complexity, The 3rd International Conference on Information Technology ICIT 2007, May 9-11, 2007 AL-Zaytoonah University, Amman, Jordan.

[4] A. Sabour, and A. Lbekkouri and M. Sabour, THE THREE STATES FUNCTIONS : FOUNDATIONS AND COMPLEXITY., ISCCSP 2008. 3rd International Symposium on Communications, Control and Signal Processing. March 12–14, 2008 at Malta Digital Object Identifier: 10.1109/ISCCSP.2008.4537342

[5] Kruskal, J.B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika, 29, 1-27, 115-129. (1964).

[6] J. B. Kruskal, M. Wish. Multidimensional Scaling. Sage Publications. Beverly Hills. CA (1977).

[7] G.L. Baker, J.P. Gollub, Chaotic Dynamics, an Introduction, 2nd Edition (Cambridge Univ. Press,Cambridge, 1996).

[8] J.P. Eckmann, D. Ruelle, Ergodic theory of chaos and strange attractors, Reviews of Modern Physics, Vol.57, N°3, Part I, 1985, pp. 622,623.

[9] R.C. Hilborn, “Chaos ans Nonlinear Dynamics”, 1994, Oxford University Press Acton, F.S., 1990, _Numerical methods that work, Corrected edition, Washington, Mathematical Association of America.

Author Profile

Abderrahim SABOUR received his PhD degree in Computer Science from the University Mohammed V-Agdal in 2007. His research interest includes artificial intelligence and computer security. He is an assistant professor at the Cadi Ayyad University since 2009. He is presently working as an Assistant Professor in the Department Of Computer Science, Higher School of Technology, Ibn Zohr University, Agadir, Morocco.